THE OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

+ HackSpace

Issue 147 | November 2024 | magpi.cc

Raspberry Pi PRESS

£7.99

Raspberry Pi AI Camera

Artificial Intelligence *Special Edition*

Incredible AI *Project Ideas*

AI Camera: *How-to Guide!*

UNUSUAL TOOLS · CNC MACHINE · LED LIGHTS

Programming
Internet
Sound & Video
Graphics
Accessories
Universal Access
Help
Preferences
Run...
Shutdown...
23:57

WELCOME
to The MagPi 147

Artificial intelligence is blazing hot at the moment. Raspberry Pi has gone from having not one, but two incredible AI products in the space of a few months. In issue 143 we revealed AI Kit; this month we're proud to cover AI Camera.

AI Camera looks like a regular Camera Module but has a Sony IMX500 Intelligent Vision Sensor that runs neural network models on the camera. It connects to Raspberry Pi using the ribbon cable, which leaves the board free to control hardware. Meanwhile, your AI Camera provides object and pose detection, and you can train AI models.

To celebrate we've gathered together some of the best AI projects from the community, and put together a how-to guide to using your new AI Camera.

Raspberry Pi is a whole lot smarter with AI Camera. Train it well!

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

This month, editor Lucy's cat has trained her to 'object detect' the more expensive cat food in the supermarket.

magpi.cc

THE OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
+ HackSpace
Raspberry Pi AI Camera
Artificial Intelligence Special Edition
Incredible AI Project ideas
AI Camera: How-to guide!
PORTABLE WORKBENCH · CNC MACHINE · LED LIGHTS

The parts we sell help us reach for the stars
DigiKey
Contents
PLAY RETRO HORROR CLASSICS ON RASPBERRY PI 5
The more things change...
HiPi.io
HIGHPI PRO
The new case from the HiPi.io team

STATEMENT OF OWNERSHIP, MANAGEMENT, AND CIRCULATION (required by Act of August 12, 1970: Section 3685, Title 39, United States Code). 1. The MagPi. 2. (ISSN 2051-9982). 3. Filing Date: September 30, 2024. 4. Issue Frequency: Monthly. 5. Number of Issues Published Annually: 12. 6. Annual Subscription Price: $80. 7. Complete Mailing Address of Known Office of Publication: Publishers Service Associates 2406 Reach Road Williamsport, PA 17701. 8. Complete Mailing Address of Headquarters or General Business Office of Publisher: Raspberry Pi 194 Science Park Cambridge, CB4 0AB UK. 9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor. Publisher: Raspberry Pi 194 Science Park Cambridge, CB4 0AB UK; Editor: Lucy Hattersley Raspberry Pi 194 Science Park Cambridge, CB4 0AB UK; Managing Editor: Lucy Hattersley Raspberry Pi 194 Science Park Cambridge, CB4 0AB UK. 10. Owner: Raspberry Pi Ltd 194 Science Park Cambridge, CB4 0AB UK. 11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities: Raspberry Pi Holdings PLC 194 Science Park Cambridge, CB4 0AB UK. 12. Tax Status: Has Not Changed During Preceding 12 Months. 13. Publication Title: The MagPi. 14. Issue Date for Circulation Data Below: September 25, 2024. 15. Extent and Nature of Circulation: A. Total Number of Copies (Net press run). Average No. Copies Each Issue During Preceding 12 Months: 2406. No. Copies of Single Issue Published Nearest to Filing Date: 2377. B. Paid Circulation. 1. Mailed Outside-County Paid Subscriptions. Average No. Copies Each Issue During Preceding 12 Months: 1535. No. Copies of Single Issue Published Nearest to Filing Date: 1510. 2. Mailed In-County Paid Subscriptions. Average No. Copies Each Issue During Preceding 12 Months: 0. No. Copies of Single Issue Published Nearest to Filing Date: 0. 3. Paid Distribution Outside the Mails Including Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Paid Distribution Outside USPS. Average No. Copies Each Issue During Preceding 12 Months: 322. No. Copies of Single Issue Published Nearest to Filing Date: 329. 4. Paid Distribution by Other Classes of Mail Through the USPS. Average No. Copies Each Issue During Preceding 12 Months: 0. No. Copies of Single Issue Published Nearest to Filing Date: 0. C. Total Paid Distribution. Average No. Copies Each Issue During Preceding 12 Months: 1857. No. Copies of Single Issue Published Nearest to Filing Date: 1839. D. Free or Nominal Rate Distribution. 1. Free or Nominal Rate Outside-County Copies. Average No. Copies Each Issue During Preceding 12 Months: 5. No. Copies of Single Issue Published Nearest to Filing Date: 6. 2. Free or Nominal Rate In-County Copies. Average No. Copies Each Issue During Preceding 12 Months: 0. No. Copies of Single Issue Published Nearest to Filing Date: 0. 3. Free or Nominal Rate Copies Mailed at Other Classes Through the USPS. Average No. Copies Each Issue During Preceding 12 Months: 0. No. Copies of Single Issue Published Nearest to Filing Date: 0. 4. Free or Nominal Rate Distribution Outside the Mail. Average No. Copies Each Issue During Preceding 12 Months: 0. No. Copies of Single Issue Published Nearest to Filing Date: 0. E. Total Free or Nominal Rate Distribution. Average No. Copies Each Issue During Preceding 12 Months: 5. No. Copies of Single Issue Published Nearest to Filing Date: 6. F. Total Distribution. Average No. Copies Each Issue During Preceding 12 Months: 1862. No. Copies of Single Issue Published Nearest to Filing Date: 1845. G. Copies not Distributed. Average No. Copies Each Issue During Preceding 12 Months: 544. No. Copies of Single Issue Published Nearest to Filing Date: 532. H. Total. Average No. Copies Each Issue During Preceding 12 Months: 2406. No. Copies of Single Issue Published Nearest to Filing Date: 2377. I. Percent Paid. Average No. Copies Each Issue During Preceding 12 Months: 99%. No. Copies of Single Issue Published Nearest to Filing Date: 99%. Publication of Statement of Ownership: will be printed in the October 24, 2024 issue of this publication. Signature and Title of Editor, Publisher, Business Manager, or Owner: Brian Jepson, Publishing Director. I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanctions (including fines and imprisonment) and/or civil sanctions (including civil penalties).

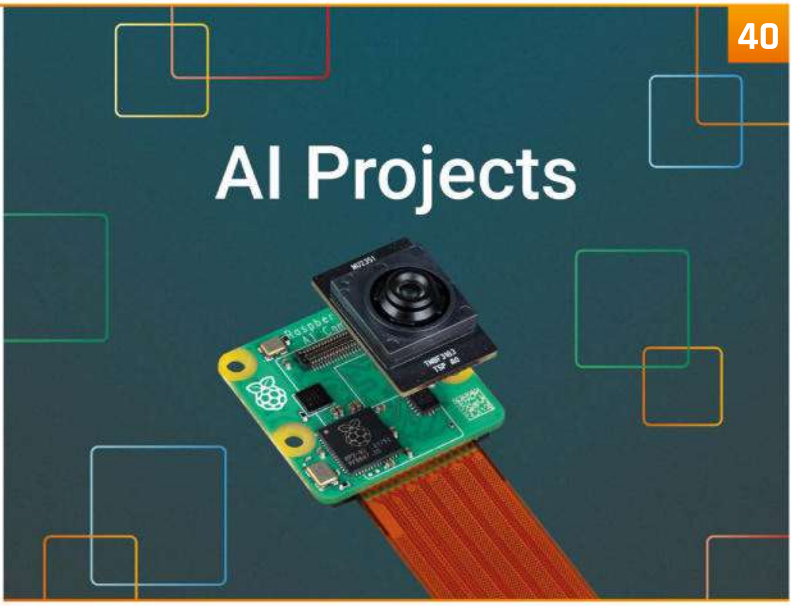
Contents

> Issue 147 > November 2024

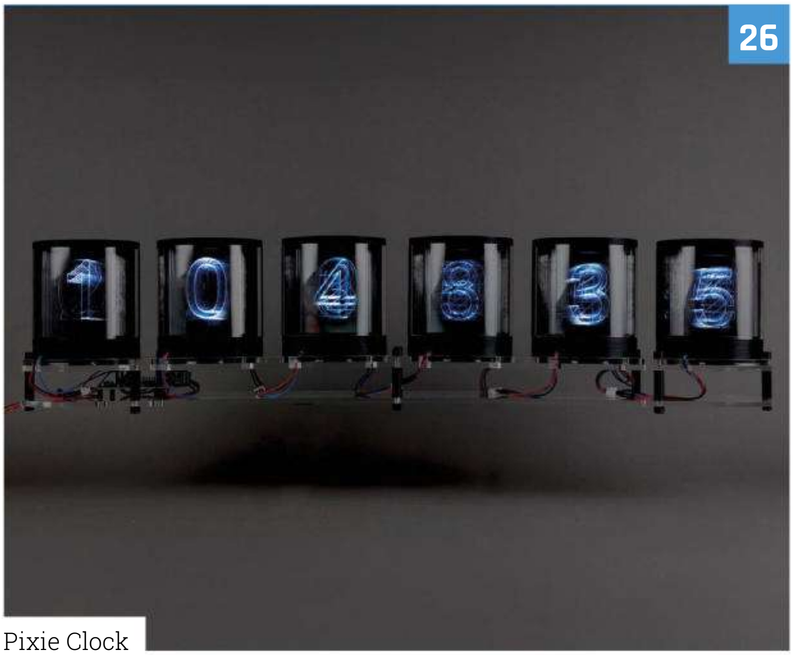
Cover Feature

Regulars

Project Showcases



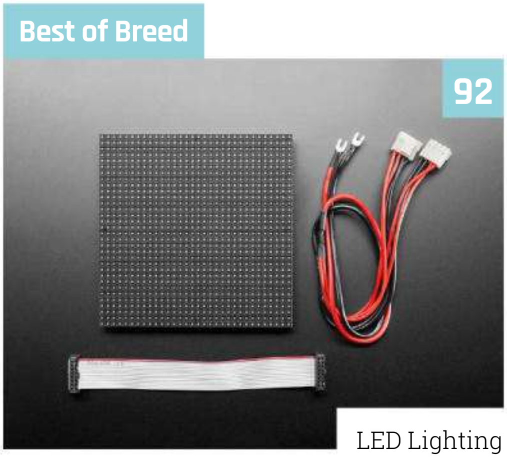
TVA TemPad



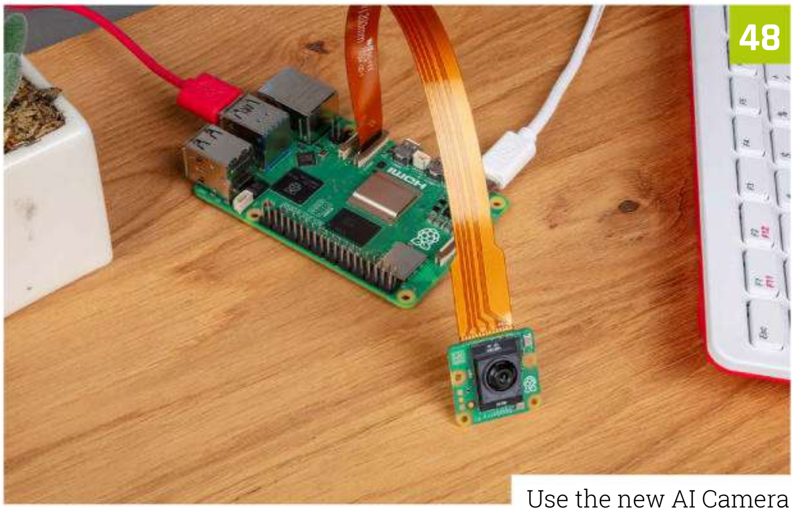
Pixie Clock

The MagPi is published monthly by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The MagPi, c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Statial.b adjustable mouse



LED Lighting



Use the new AI Camera

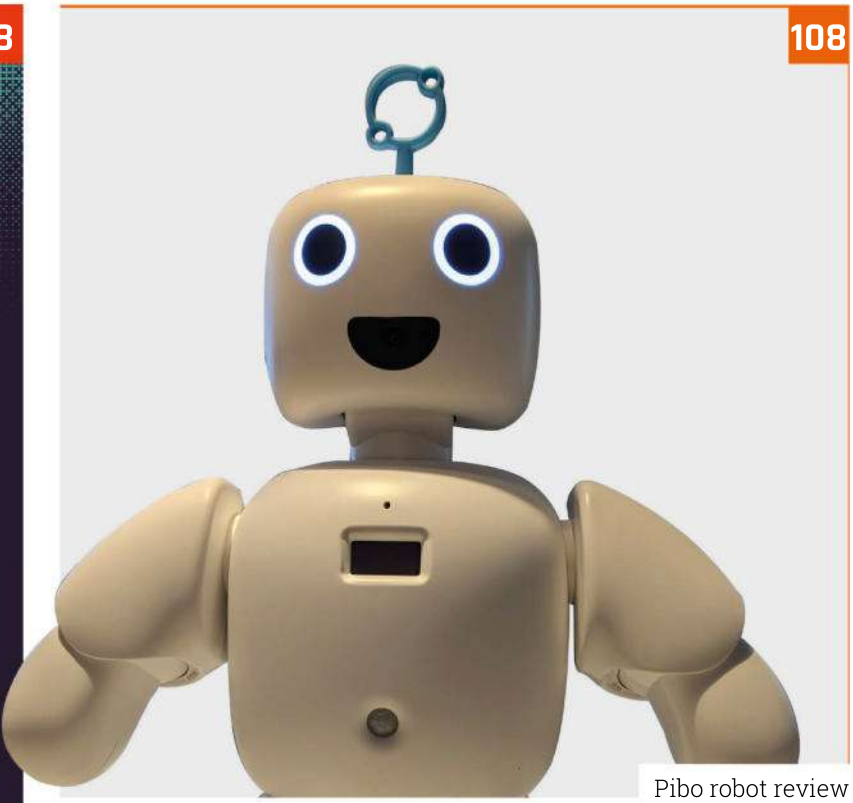
Robot Kalimba

Unusual tools: steel cable tie tensioning tool

GSS
OPTION
ams
NXP
HARWIN
PANDUIT
Honeywell
DELTA
Amphenol
MICROCHIP
PHOENIX CONTACT
aeris
muRata
molex
ST
onsemi
TE
ANALOG DEVICES
infineon
AMD
omron
KEMET
SILICON LABS
TDK
Adesto
CEL
DigiKey
we get technical

The Big Feature

Handheld console gaming



Pibo robot review

Reviews

Community

Kari Lawler interview

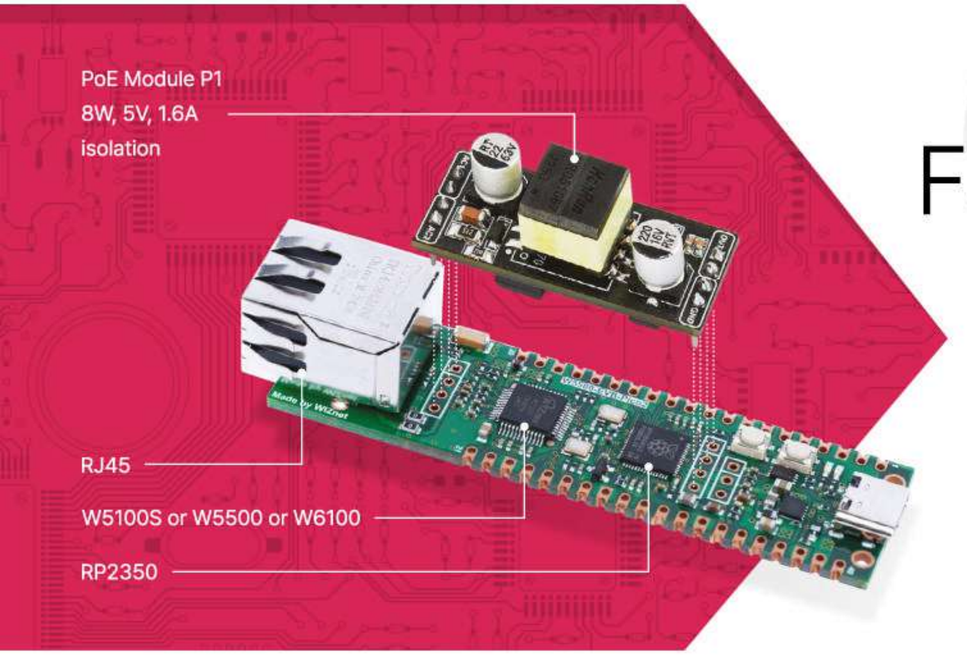
WIN 1 OF 5 RASPBERRY PI **AI CAMERA** 126

DISCLAIMER: Some of the tools and techniques shown in *The MagPi* magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in *The MagPi* magazine. Laws and regulations covering many of the topics in *The MagPi* magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in *The MagPi* magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

WIZnet

Ethernet connectivity solution for **RP2350**

EVB-Pico2
Family boards



Features

Available in W5100S, W5500 or W6100 version

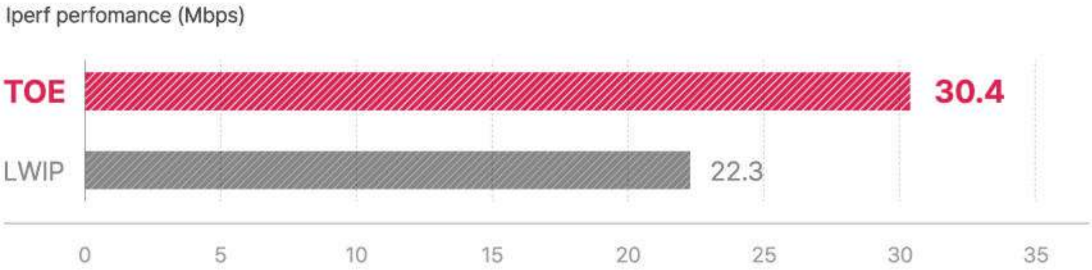
PoE enabled via add-on module

Identical pinout with Raspberry Pi Pico

USB Type C

Run button available

TOE vs LWIP comparison



Tests were done using RP2350 (150Mhz) and W6100 (SPI set to 37.5Mhz)

PoE Module specifications

WIZPoE - S1

IEEE802.3af compliant
Mode A(Endspan), Mode B(Midspan)
Wide input voltage range 40Vdc ~ 60Vdc
High DC/DC conversion efficiency

Non-Isolation
Internal build in 2 channel bridge rectifiers
5V/8W Output

WIZPoE - P1

IEEE802.3af compliant
Mode A(Endspan), Mode B(Midspan)
Wide input voltage range 40Vdc ~ 60Vdc
High DC/DC conversion efficiency

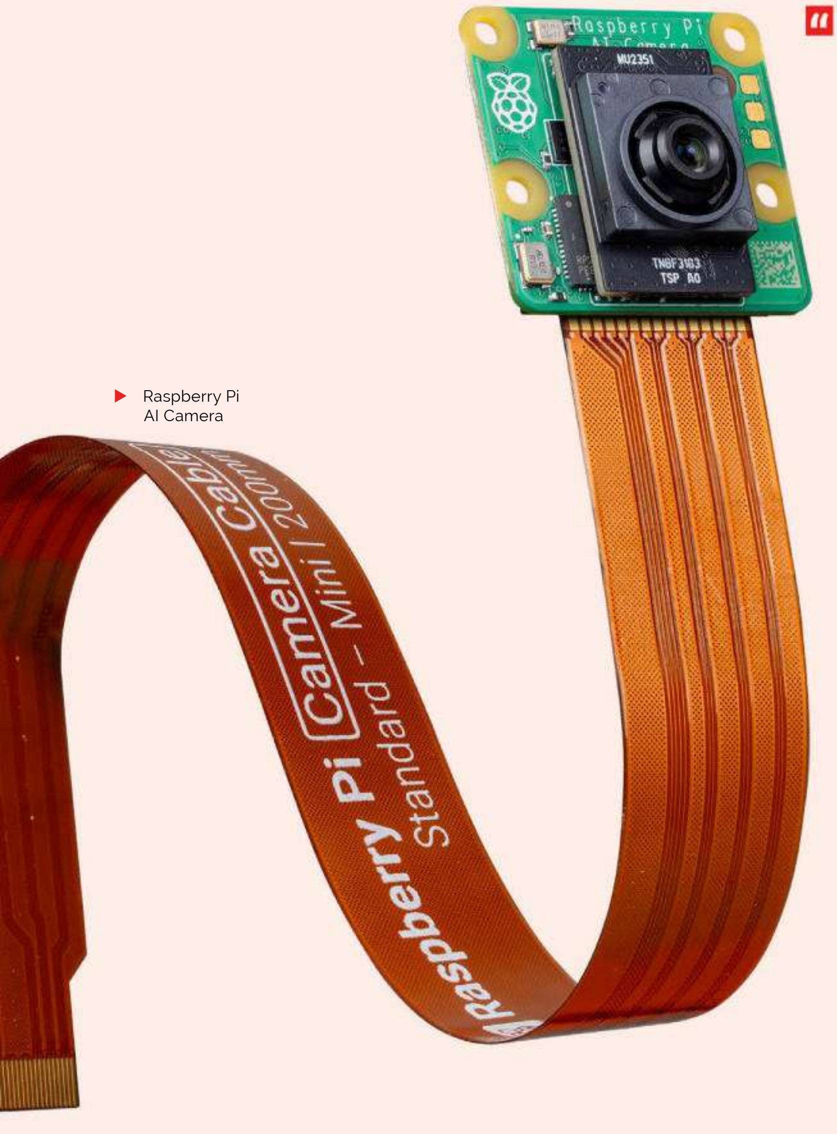
Isolation
Internal build in 2 channel bridge rectifiers
5V/8W Output

For reselling & bulk purchases, please contact direct@wiznet.io

More info

Raspberry Pi AI Camera

Camera Module with built-in artificial intelligence on sale now.
By **Naush Patuck**



Raspberry Pi AI Camera

> The AI Camera is built around a Sony IMX500 image sensor with an integrated AI accelerator

People have been using Raspberry Pi products to build artificial intelligence projects for almost as long as we've been making them. As we've released progressively more powerful devices, the range of applications that we can support natively has increased; but in any generation there will always be some workloads that require an external accelerator, like the Raspberry Pi AI Kit, which we launched in June (**magpi.cc/aikit**).

The AI Kit is an awesomely powerful piece of hardware, capable of performing 13 trillion operations per second. But it is only compatible with Raspberry Pi 5, and requires a separate camera module to capture visual data. We are very excited therefore to announce a new addition to our camera product line: the Raspberry Pi AI Camera (**magpi.cc/aicamera**).

The AI Camera is built around a Sony IMX500 image sensor with an integrated AI accelerator. It can run a wide variety of popular neural network models, with low power consumption and low latency, leaving the processor in your Raspberry Pi free to perform other tasks.

Key features of the Raspberry Pi AI Camera include:

- 12 MP Sony IMX500 Intelligent Vision Sensor
- Sensor modes: 4056×3040 at 10fps, 2028×1520 at 30fps
- 1.55 µm × 1.55 µm cell size
- 78-degree field of view with manually adjustable focus
- Integrated RP2040 for neural network and firmware management

The AI Camera can be connected to all Raspberry Pi models, including Raspberry Pi Zero, using our regular camera ribbon cables.

Using Sony's suite of AI tools, existing neural network models using frameworks such as TensorFlow or PyTorch can be converted to run efficiently on the AI Camera. Alternatively, new models can be designed to take advantage of the AI accelerator's specific capabilities.

Under the hood

To make use of the integrated AI accelerator, we must first upload a model. On older Raspberry Pi devices this process uses the I2C protocol, while on Raspberry Pi 5 we are able to use a much faster custom two-wire protocol. The camera end of the link is managed by an on-board RP2040 microcontroller; an attached 16MB flash device caches recently used models, allowing us to skip the upload step in many cases.

Once the sensor has started streaming, the IMX500 operates as a standard Bayer image sensor, much like the one on Raspberry Pi Camera Module 3. An integrated image signal processor (ISP) performs basic image processing steps on the sensor frame (principally Bayer-to-RGB conversion and cropping/rescaling), and feeds the processed frame directly into the AI accelerator. Once the neural network model has processed the frame, its output is transferred to the host Raspberry Pi together with the Bayer frame over the CSI-2 camera bus.

Integration with Raspberry Pi libcamera

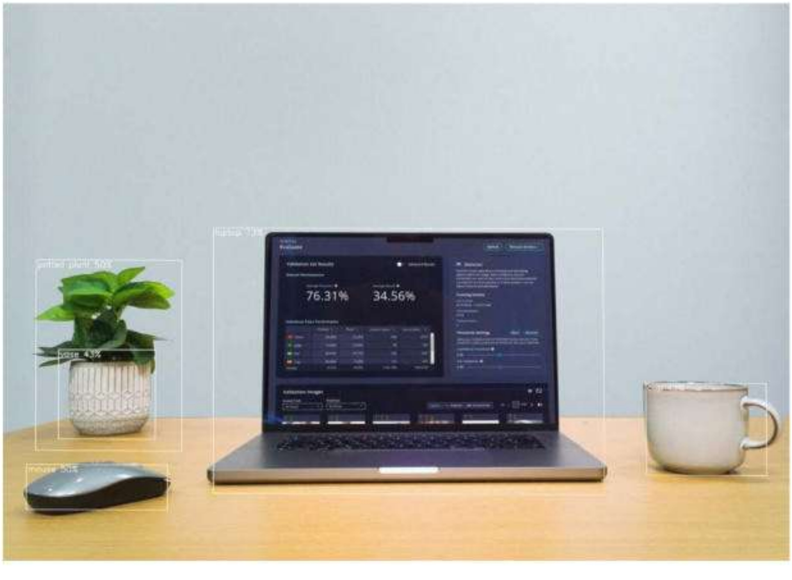
A key benefit of the AI Camera is its seamless integration with our Raspberry Pi camera software stack. Under the hood, libcamera processes the Bayer frame using our own ISP, just as it would for any sensor.

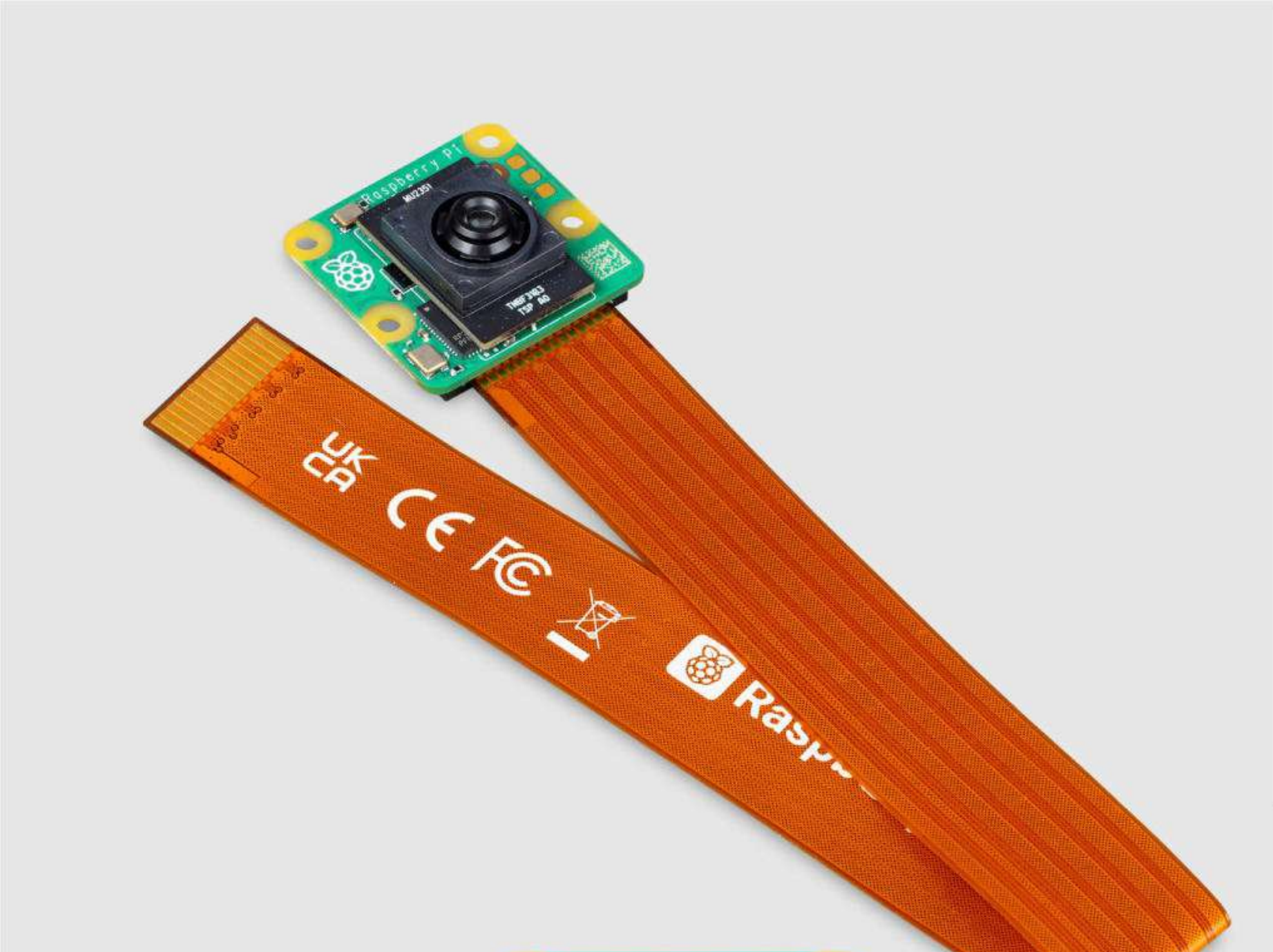
We also parse the neural network results to generate an output tensor, and synchronise it with the processed Bayer frame. Both of these are returned to the application during libcamera's request completion step.

The Raspberry Pi camera frameworks – Picamera2 and rpicam-apps, and indeed any libcamera-based application – can retrieve the output tensor, correctly synchronised with the sensor frame. Here's an example of an object detection neural network model (MobileNet SSD) running under rpicam-apps and performing inference on a 1080p video at 30fps.

This demo uses the post-processing framework in rpicam-apps to generate object bounding boxes from the output tensor and draw them on the

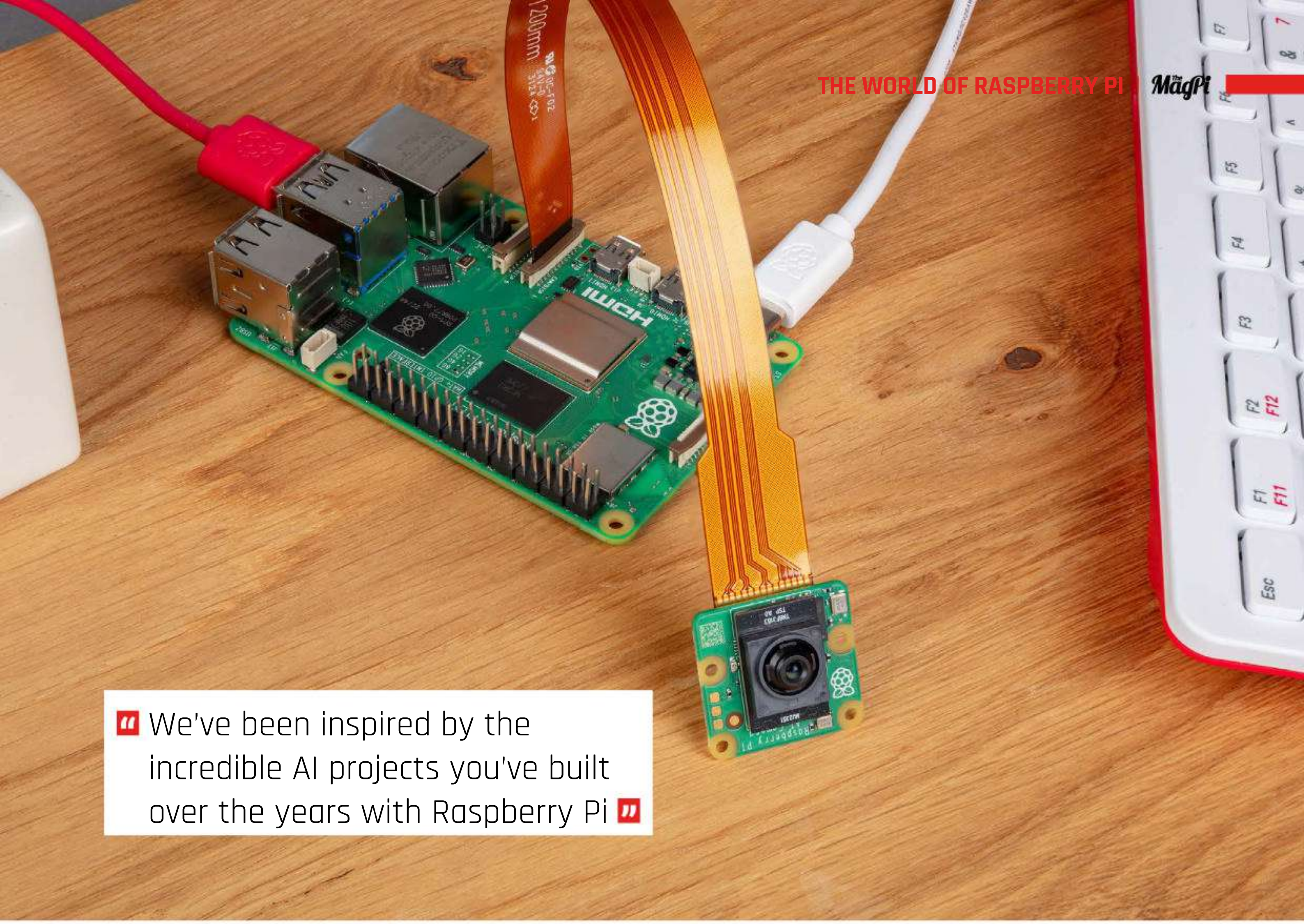
▼ Object detection with bounding boxes





▲ AI Camera connects to Raspberry Pi using the standard cable connector

▶ The Camera features a IMX500 sensor



> We've been inspired by the incredible AI projects you've built over the years with Raspberry Pi

▲ AI Camera attached to Raspberry Pi 5

image. This stage takes no more than 300 lines of code to implement. An equivalent application built using Python and Picamera2 requires many fewer lines of code.

Another example below shows a pose estimation neural network model (PoseNet) performing inference on a 1080p video at 30fps.

Although these examples were recorded using a Raspberry Pi 4, they run with the same inferencing performance on a Raspberry Pi Zero!

Together with Sony, we have released a number of popular visual neural network models optimised for the AI Camera in our model zoo, along with visualisation example scripts using Picamera2.

Which product should I buy?

Should you buy a Raspberry Pi AI Kit, or a Raspberry Pi AI Camera? The AI Kit has higher theoretical performance than the AI Camera, and can support a broader range of models, but is only compatible with Raspberry Pi 5. The AI Camera is more compact, has a lower total cost if you don't already own a camera, and is compatible with all models of Raspberry Pi.

Ultimately, both products provide great acceleration performance for common models, and both have been optimised to work smoothly with our camera software stack.

Getting started and going further

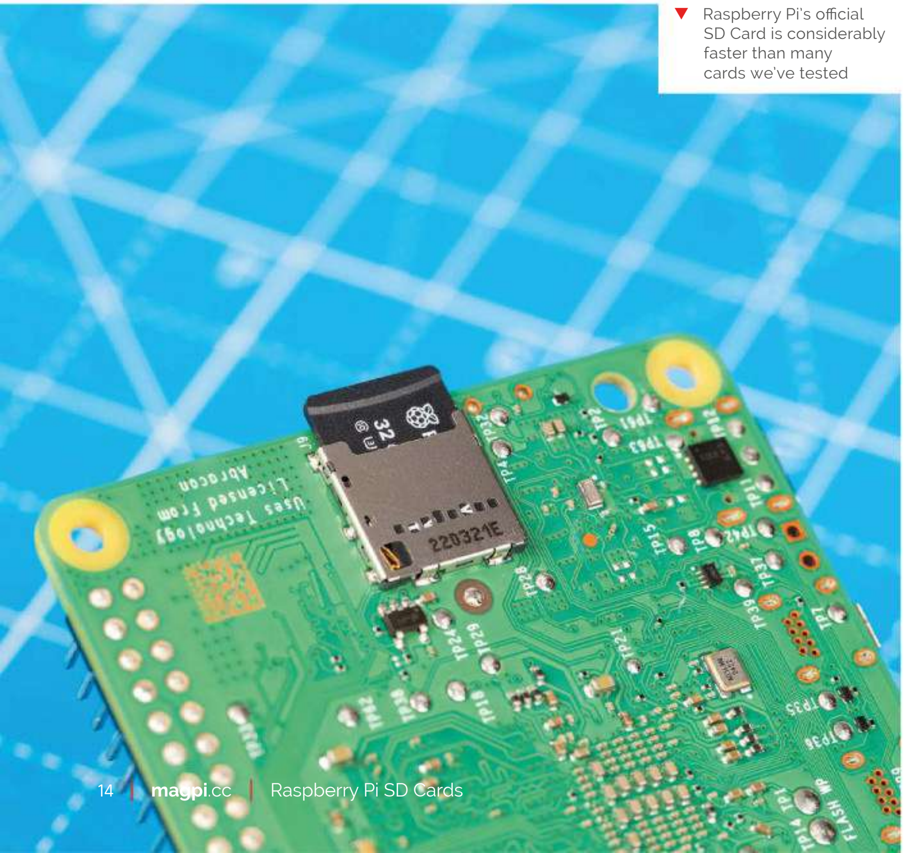
Check out our getting started tutorial (page 48). There you'll find instructions on installing the AI Camera hardware, setting up the software environment, and running the examples and neural networks in our model zoo (**magpi.cc/imx500models**).

Sony's AITRIOS Developer (**magpi.cc/devimx500**) site has more technical resources on the IMX500 sensor, in particular the IMX500 Converter (**magpi.cc/imx500conv**) and IMX500 Package (**magpi.cc/imx500pkg**) documentation, which will be useful for users who want to run custom-trained networks on the AI Camera.

We've been inspired by the incredible AI projects you've built over the years with Raspberry Pi, and your hard work and inventiveness encourages us to invest in the tools that will help you go further. The arrival of first the AI Kit, and now the AI Camera, opens up a whole new world of opportunities for high-resolution, high-frame rate, high-quality visual AI: we don't know what you're going to build with them, but we're sure it will be awesome.

Raspberry Pi **SD Cards**

Official A2-class microSD cards are now available.
By **Jonathan Bell**



Raspberry Pi's official SD Card is considerably faster than many cards we've tested

With the latest release of Raspberry Pi OS, Raspberry Pi 5 can make use of the extra performance available from Class A2 microSD cards. And to help you take advantage of this, we are introducing our own range of high-quality, low-cost Raspberry Pi SD Cards.

Cards on the table

As many of you will know first-hand, your choice of microSD card makes a huge difference to your Raspberry Pi experience. Historically, we've worked with our Approved Reseller partners to test and endorse third-party microSD cards. But as cards have become more sophisticated, and particularly with the advent of Class A2 cards, this process has become increasingly cumbersome.

To ensure you have the best possible experience at the lowest possible cost, we've worked with our partner Longsys to develop a range of branded Raspberry Pi SD Cards (**magpi.cc/sdcard**). These Class A2 cards offer exceptional random read and write throughput across the entire range of Raspberry Pi computers, and when used on Raspberry Pi 5 support command queueing for even higher performance.

From today, our Approved Resellers will only promote Raspberry Pi SD Cards alongside Raspberry Pi computers, and you can be assured of their quality.

Top of the class

Cards which support Application Performance Class A2, such as our new Raspberry Pi SD Cards, enable faster read and write operations, and Raspberry Pi 5 incorporates hardware features which allow it to make the most of this extra performance. To enable these features, you will need to use the latest release of Raspberry Pi OS, or update your Raspberry Pi OS install with the latest packages. Run the following command to update:

```
sudo apt update && sudo apt full-upgrade
```

How exactly do Class A2 cards achieve better performance? Read on!

What is CQHCI?

The SD Host Controller Interface (SDHCI) specification standardises the piece of hardware (the host controller) which controls communication with the SD card. On Raspberry Pi computers, the host controller lives inside the Broadcom application processor. The Command Queueing Host Controller Interface (CQHCI) extends SDHCI with an extra set of control registers, and a CQ engine which takes over from the legacy host controller when a suitable card is detected.

Cards must be explicitly put into command queueing (CQ) mode, after which a new set of SD commands becomes available and many of the existing SD commands become invalid. The new commands decouple the request to read or write a card sector from the response of the card. Each read or write operation is tagged, with up to 32 tags in use across both reads and writes. The card can choose the order in which it returns responses to the commands, and may optionally buffer write data rather than committing it immediately to flash.

> “Each read or write operation is tagged, with up to 32 tags in use”

By allowing it to effectively “see into the future”, command queueing lets the flash controller hide more of the latency associated with accessing disparate NAND flash pages. This results – at least in theory – in better throughput for random I/O workloads of the sort generated by Raspberry Pi OS.

CQ support first landed in eMMC devices with JEDEC specification JESD84-B51, in 2015. The SD specification equivalent landed some time later with SD v6.00, in 2017. However, at the time of the Raspberry Pi 5 launch in 2023, Linux only supported CQHCI on eMMC devices – so we were leaving performance on the table.

In early 2024 I set about implementing the missing CQ support for SD cards.

How do you use CQHCI?

Carefully parsing the SD specification led me to develop a dependency chain of optional card features that all needed to be supported if CQ mode is to be used. These are, in order:

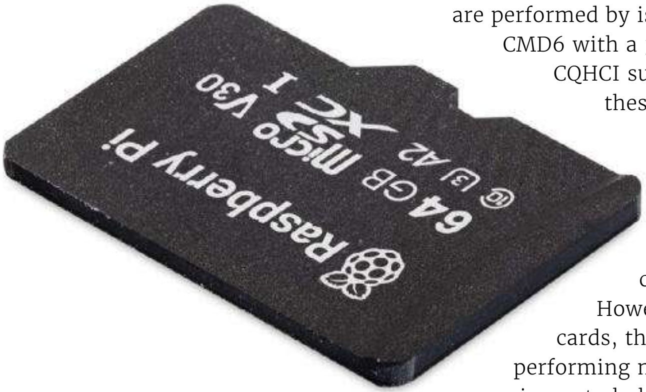
- The card must support Extension Register access, which is a generic method of accessing optional features over 512-byte pages, each with a type identifying to what feature extension the page refers
- The card must support the Performance Enhancement extension registers
- In the Performance Enhancement extension, the card must support Write Caching
- As a consequence of Write Caching support, the card must also support the Power extension registers and at a minimum support Power-Off notifications
- The card must declare the queue depth required to meet Class A2 performance – from two to 32 tags

> While best performance on Raspberry Pi 5 was our primary goal, the non-CQ performance of these cards is still stonkingly fast

As Linux already supported CQ with eMMC cards, all I had to do was to find out where the SD implementation differed – and there were a few of these cases.

During normal operation the host operating system sometimes needs to issue 'meta-ops' that don't directly transfer data but do related things, such as recalibrating the host-to-card data path delays, requesting card status as a proxy for card removal, and doing flash maintenance operations such as signalling block discard.

For eMMC devices, most meta-ops are performed by issuing command CMD6 with a 32bit argument. CQHCI supports injecting these while in CQ mode by designating the 'top' tag in the controller for performing DCMDs (direct commands). However, with microSD cards, the set of commands performing meta-ops generally require us to halt the CQ engine, and issue a non-CQ command using the regular SD host controller registers.



Once these differences were ironed out, I had a workable Linux driver, which was pushed to rpi-update. I created a testing thread in the forums (**magpi.cc/testingsdcards**) for the adventurous, and set about evaluating my extensive collection of retail cards.

Testing cards

How well do microSD cards implement CQ mode? In a very hit-and-miss fashion.

SanDisk cards, in particular the Extreme and Extreme Pro product lines, were my first choice – and they performed well. However, other manufacturers' offerings suffered from one or more of a number of common deficiencies that precluded CQ mode operation, or caused them to flake out in use:

- Not declaring Power-Off notification support despite implementing the extension
- Hanging on receipt of a cache flush request after CQ mode had been activated then deactivated
- Cards not correctly implementing the 'CQ enable' expansion register bit – if I wrote a 1, I would still read back 0 forever

There was even one type of card that claimed Class A2 support but ignored any request to read the expansion registers to probe for any of these features!

The Raspberry Pi kernel filters out cards that fail these tests, either during feature probing or with an explicit quirk that matches the card identifier. If you find an A2-branded card that misbehaves on a Raspberry Pi 5, then please report it in the above-mentioned forum thread.

Write caching + surprise removal = bad

One potential pitfall of enabling CQ mode is that it provides cards with new opportunities to corrupt your filesystem if power is removed unexpectedly. In CQ mode, hosts should honour the requirement to maintain the card's power supply, and only remove it after a Power-Off notification is sent; this provides an opportunity for the flash controller to commit all outstanding writes to flash. For battery-powered hosts with concealed microSD slots such as a phone, that is an easy contract to fulfil – requesting device shutdown or uncovering the slot can trigger a Power-Off notification.

Raspberry Pi, with its exposed microSD slot and pluggable PSU, has a harder time providing this guarantee.

With multiple writes in flight, or multiple posted notifications of pending writes, we can no longer guarantee the order in which writes get committed to flash. If power is removed unexpectedly, an arbitrary collection of recent writes may not have been committed, rather than strictly the n most recent writes; this greatly complicates the task of making the filesystem resilient to corruption. The Raspberry Pi kernel sidesteps this problem by limiting the maximum number of posted writes in CQ mode to one. While in theory this may result in lower sequential write throughput, the cards I've tested see at most a 2-3% percent reduction in performance.

Introducing Longsys

Once it became clear that Class A2 microSD cards offer a significant performance uplift when operating in CQ mode on Raspberry Pi 5, we started discussions with several card OEMs, with the goal of qualifying a cost-effective offering that would work well across every generation of Raspberry Pi.

We settled on Longsys (**longsys.com**) as our vendor after working with their engineering team to align their cards' declared feature sets with our requirements; to prove that the cards were robust by automatically performing over 100,000 surprise power cycles under I/O heavy load; and to tune the cards to get the best out of Raspberry Pi 5.

While best performance on Raspberry Pi 5 was our primary goal, the non-CQ performance of these cards is still stonkingly fast, and you will generally see a significant uplift in performance on older Raspberry Pi computers.

◄ The cards are available in 32, 64, and 128GB capacities

▼ Raspberry Pi's new Bumper and SD Card

Raspberry Pi Bumper for Raspberry Pi 5

More excellent new accessories. By **Jonathan Bell**

Silicon protection for your Raspberry Pi

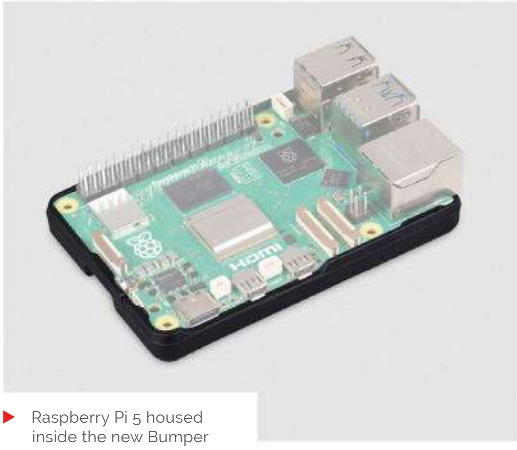
Another great recent accessory launch is the Raspberry Pi Bumper: the simple casing solution you never knew you needed, and already a firm favourite here at Pi Towers. It's a snap-on silicone base that unfussily protects the base and edges of your Raspberry Pi 5, and the surface you're putting it down on, and also makes it easy to use the power button. It's compatible with the Raspberry Pi Active Cooler, and will set you back a meagre £2.80/$3.

Here's what you get:

- One-piece flexible silicone rubber bumper
- Enables easy access to the power button
- Mounting holes remain accessible underneath the bumper

The Raspberry Pi Bumper will help protect and secure your Raspberry Pi 5.Find out more at **magpi.cc/bumper**.

> A snap-on silicone base that unfussily protects the base and edges of your Raspberry Pi 5



Raspberry Pi 5 housed inside the new Bumper

$ 79.99
PIRONMAN 5
Pironman 5 NVMe PIP

Argon ONE
V5

Argon
neo 5 NVMe

Argon
Poly+5

Automatic pressure cleaner

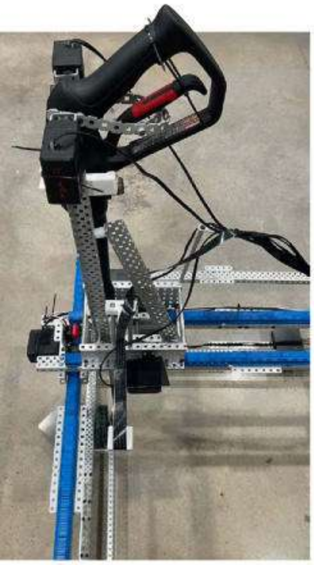
A robotic pressure washer can analyse and clean areas of grime with thanks to a Raspberry Pi. **Rob Zwetsloot** takes a splash

MAKER

Connor

A Miami high-schooler and YouTuber, Connor competes in robotics competitions and also runs cross country.

magpi.cc/garagerobots



The APC pulls the trigger on the pressure washer wand in a very simple solution to activating it

We've all been there. You see a tough dirty spot and think 'ah, if only a robot could do it for me'. Or is that just us? Well, the thought did occur to Connor at least:

"I first thought of [my project] because of a dirt patch on the concrete in my backyard," Connor tells us. "Initially, I wanted to just use a regular old pressure cleaner, but then I thought how much easier and less time-consuming it would be if I could relax and watch a robot do the job for me."

Over the course of a few months, Connor built that robot, a device to automatically clean a space measuring two feet by three feet (about 610mm × 915mm in new money).

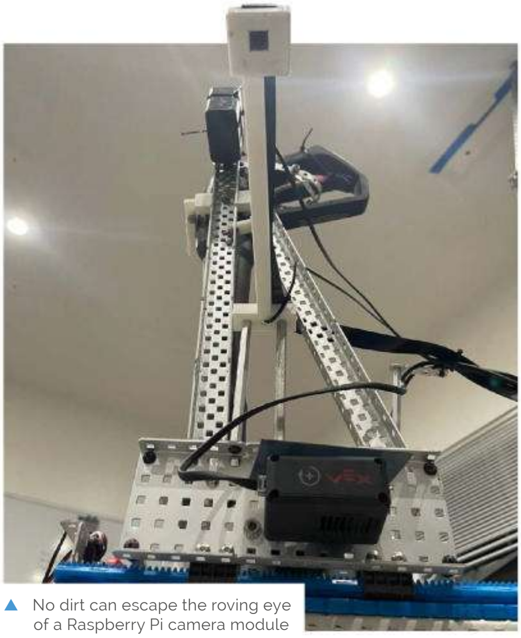
"I almost ran out of dirt to clean by the end of it and realised it may have been less effort to just go ahead and clean it," admits Connor.

Dirt vision

The system works by comparing before and after images of the spot it's trying to clean.

"The APC has a slot for the pressure cleaner, and a camera lies atop this slot," Connor explains. "The machine uses a Raspberry Pi camera module operated by Raspberry Pi to take pictures and convert them to greyscale. The program takes an average of the brightness of all the pixels. Then, under Raspberry Pi's command, the VEX motors move the pressure nozzle in a grid-like fashion while engaging the pressure cleaner's trigger. The whole cleaning process takes about a minute and a half to complete. After that, the camera and nozzle are reset to their original position, and another picture is taken and converted to greyscale. Raspberry Pi calculates the subsequent average pixel brightness. If there is a significant enough increase in brightness, the APC moves on to the next section."

Connor chose a Raspberry Pi and a camera module for this project because the image analysis was a lot simpler than some more advanced neural network implementations.



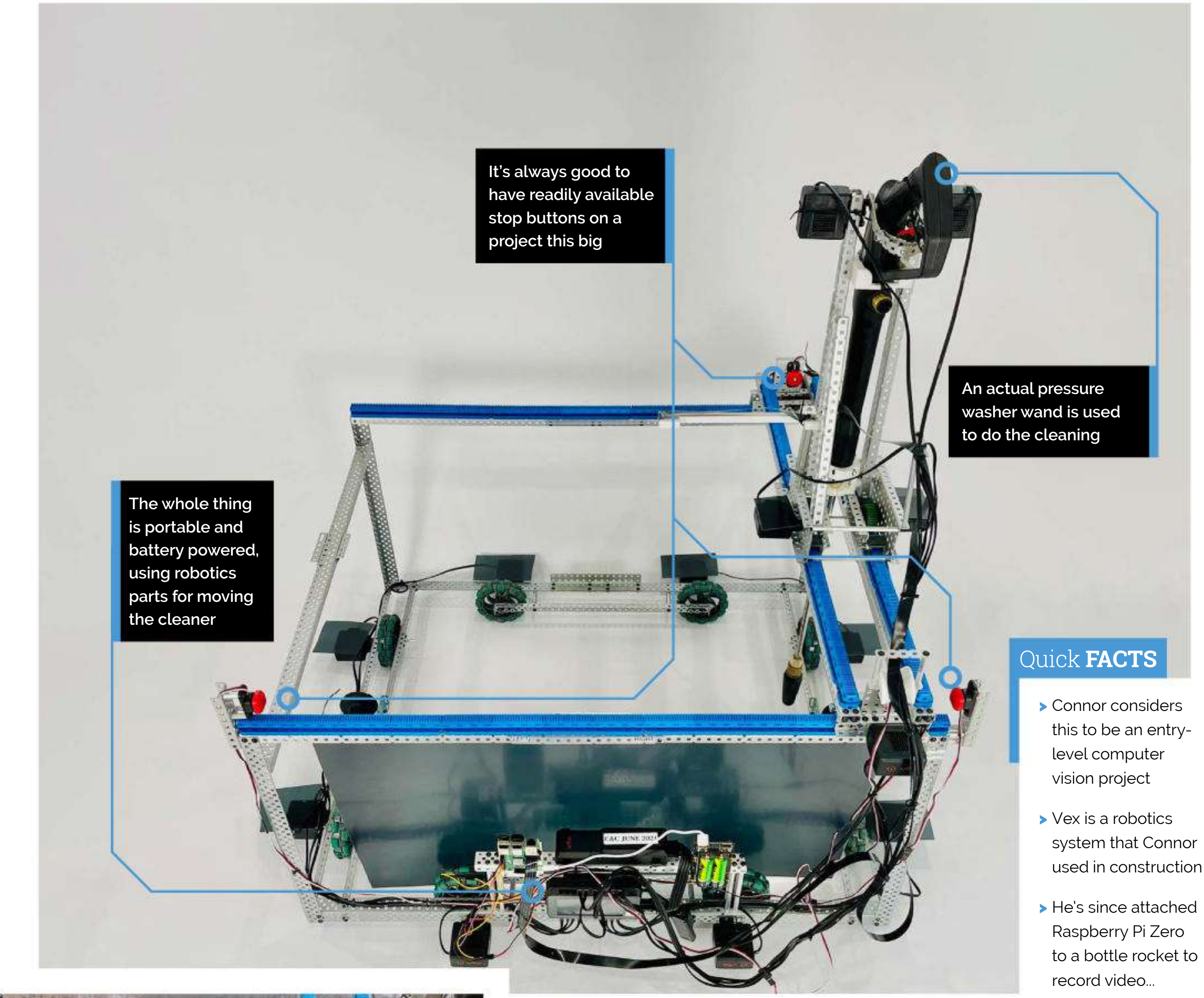
No dirt can escape the roving eye of a Raspberry Pi camera module

All's well

Getting it completed was a bit of a challenge though. "The build process was … a learning experience," Connor says. "The most straightforward part was obtaining and analysing the image. The physical construct proved more challenging. The wheels slipped on the wet surface, so weights needed to be spread along the base judiciously. The moving gantry ripped off the wires, so [cable] ties, tape, and rubber bands kept the many moving wires clear of snagging."

In a spot of good luck, Connors school had a 'Shark Tank Pitch Competition' a month after he finished it. *Shark Tank* is the US name for the TV show *Dragon's Den*, and he was able to use it as his presentation.

"Moving the entire machine to the school required a bit of deconstruction and reconstruction, but the effort was worth it," Connor tells us. "The reaction was both praise for its uniqueness and an acknowledgment that more progress was needed for viable commercialisation."



It's always good to have readily available stop buttons on a project this big

An actual pressure washer wand is used to do the cleaning

The whole thing is portable and battery powered, using robotics parts for moving the cleaner

Quick FACTS

- Connor considers this to be an entry-level computer vision project
- Vex is a robotics system that Connor used in construction
- He's since attached Raspberry Pi Zero to a bottle rocket to record video…
- …in which an Einstein LEGO minifig went up to the Karman line and back
- His next project involves identifying buildings on radiographs with neural networks

> The most straightforward part was obtaining and analysing the image. The physical construct proved more challenging

Despite how far-fetched it seems, it completely works

DEC Flip-Chip tester

Rebuilding an old PDP-9 computer with a Raspberry Pi-based device that tests hundreds of components. By **Rosie Hattersley**

MAKER

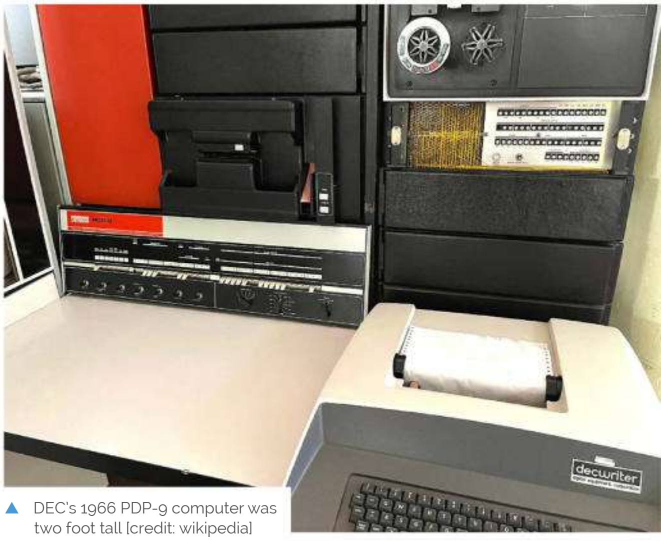
Anders Sandahl

Swedish embedded systems engineer Anders uses Linux daily and enjoys restoring gadgets and old computers to working order.

magpi.cc/brstester

Anders Sandahl loves collecting old computers: "I really like to restore them and get them going again." For this project he wanted to build a kind of a component tester for old DEC (Digital Equipment Corporation) Flip-Chip boards before he embarked on the lengthy task of restoring his 1966 PDP-9 computer – a two-foot-tall machine with six to seven hundred Flip-Chip boards inside – back to working order.

His Raspberry Pi-controlled DEC Flip-Chip tester checks the power output of these boards using relay modules and signal clips, giving accurate information about each one's power draw and output. Once he's confident each component is working properly, Anders can begin to assemble the historic DEC PDP-9 computer, which Wikipedia advises is one of only 445 ever produced: **magpi.cc/pdp9**.



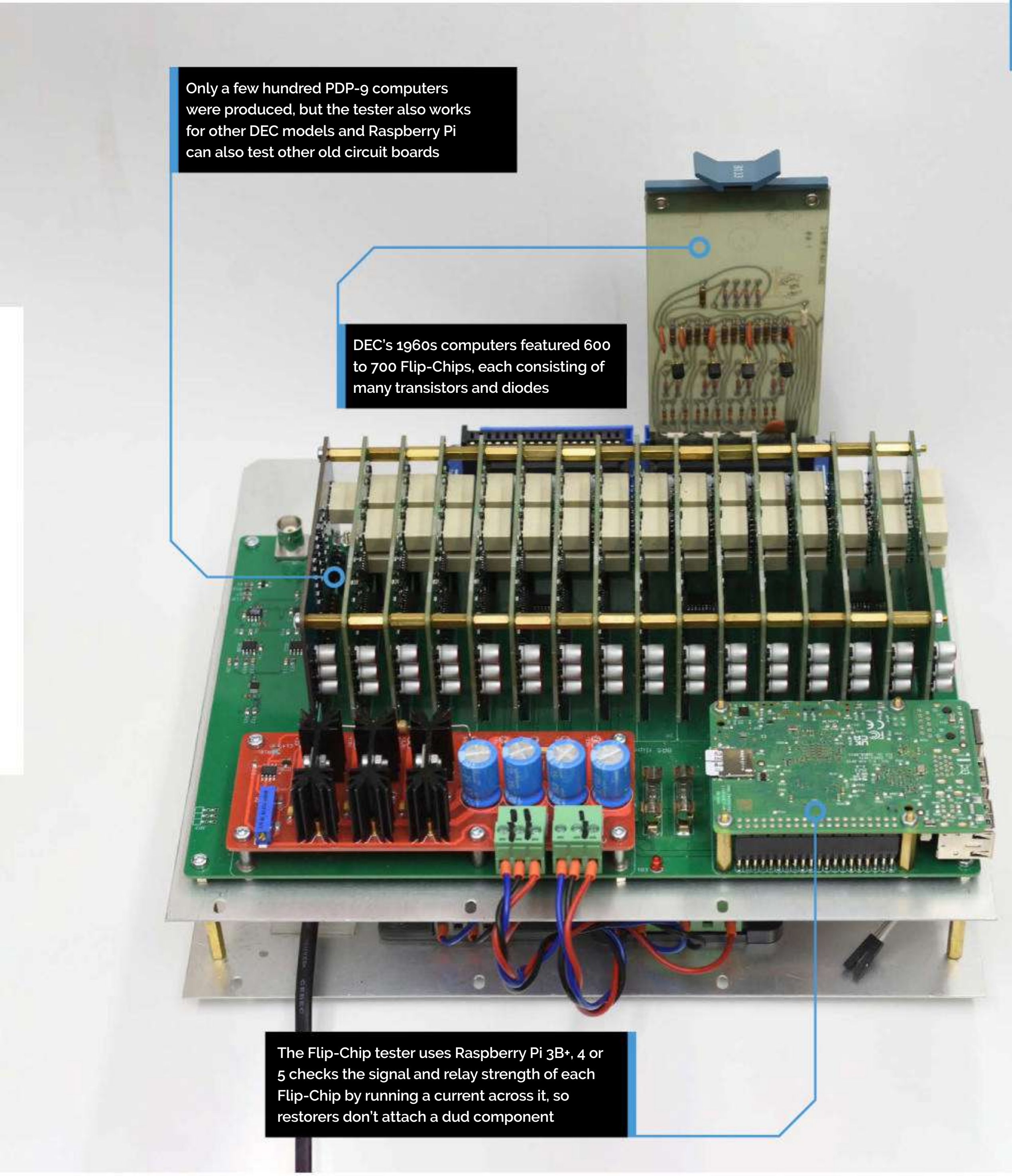
▲ DEC's 1966 PDP-9 computer was two foot tall [credit: wikipedia]

Logical approach

"Flip-Chip boards from this era implement simple logical functions, comparable to one 7400-series logic circuit," Anders explains (**magpi.cc/7400series**). "The tester uses Raspberry Pi and an ADC (analogue to digital converter) to measure and control analogue signals sent to the Flip-Chip, and digital signals used to control the tester's circuits. PDP-7, PDP-8 (both 8/S and Straight-8), PDP-9 and PDP-10 (with the original KA processor) all use this generation of Flip-Chips. A testing device for one will work for all of them, which is pretty useful if you're in the business of restoring old computers.

Rhode Island Computer Museum (RICM) is where *The MagPi* publisher Brian Jepson and friend Mike Thompson both volunteer. Mike is part of a 12-year-project to rebuild RICM's own DEC PDP-9 and, after working on a different Flip-Chip tester there (**magpi.cc/fctester**), got in touch with Anders about his Raspberry Pi-based version. He's now busily helping write the user manual for the tester unit.

Mike explains: "Testing early transistor-only Flip-Chips is incredibly complicated because the voltages are all negative, and the Flip-Chips must be tested with varying input voltages and different loads on the outputs." There are no integrated circuits, just discrete transistors. Getting such an old computer running again is "quite a task" because of the sheer number of broken components on each PCB, and Flip-Chip boards holding lots of transistors and diodes "all of which are subject to failure after 55+ years".



Only a few hundred PDP-9 computers were produced, but the tester also works for other DEC models and Raspberry Pi can also test other old circuit boards

DEC's 1960s computers featured 600 to 700 Flip-Chips, each consisting of many transistors and diodes

The Flip-Chip tester uses Raspberry Pi 3B+, 4 or 5 checks the signal and relay strength of each Flip-Chip by running a current across it, so restorers don't attach a dud component

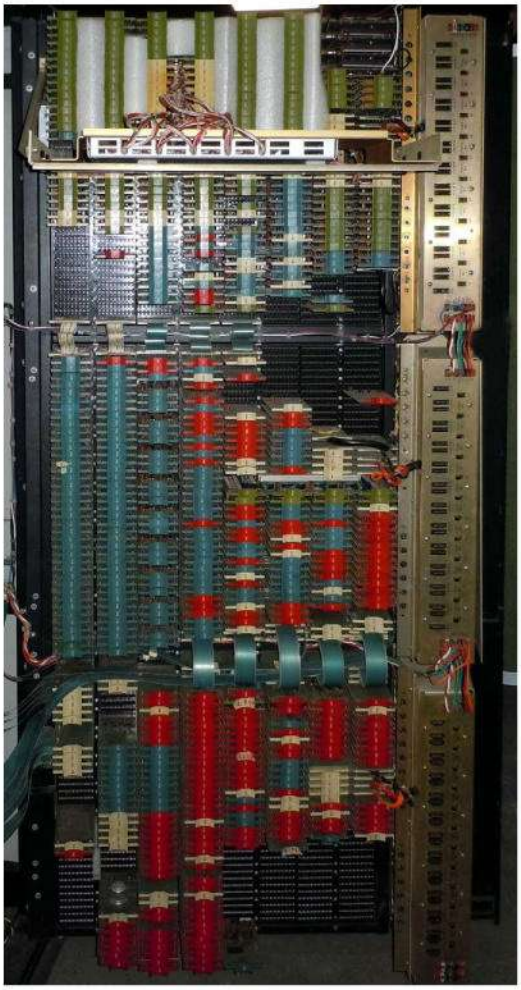
Quick FACTS

- The Flip-Chip tester has 62 relays
- Testing PCBs sounds like using an old telephone switchboard
- PDP stands for programmed data processing
- Anders previously used Raspberry Pi to build a replica PDP-8
- Take a look at the PiDP-8 at **magpi.cc/pidp8**

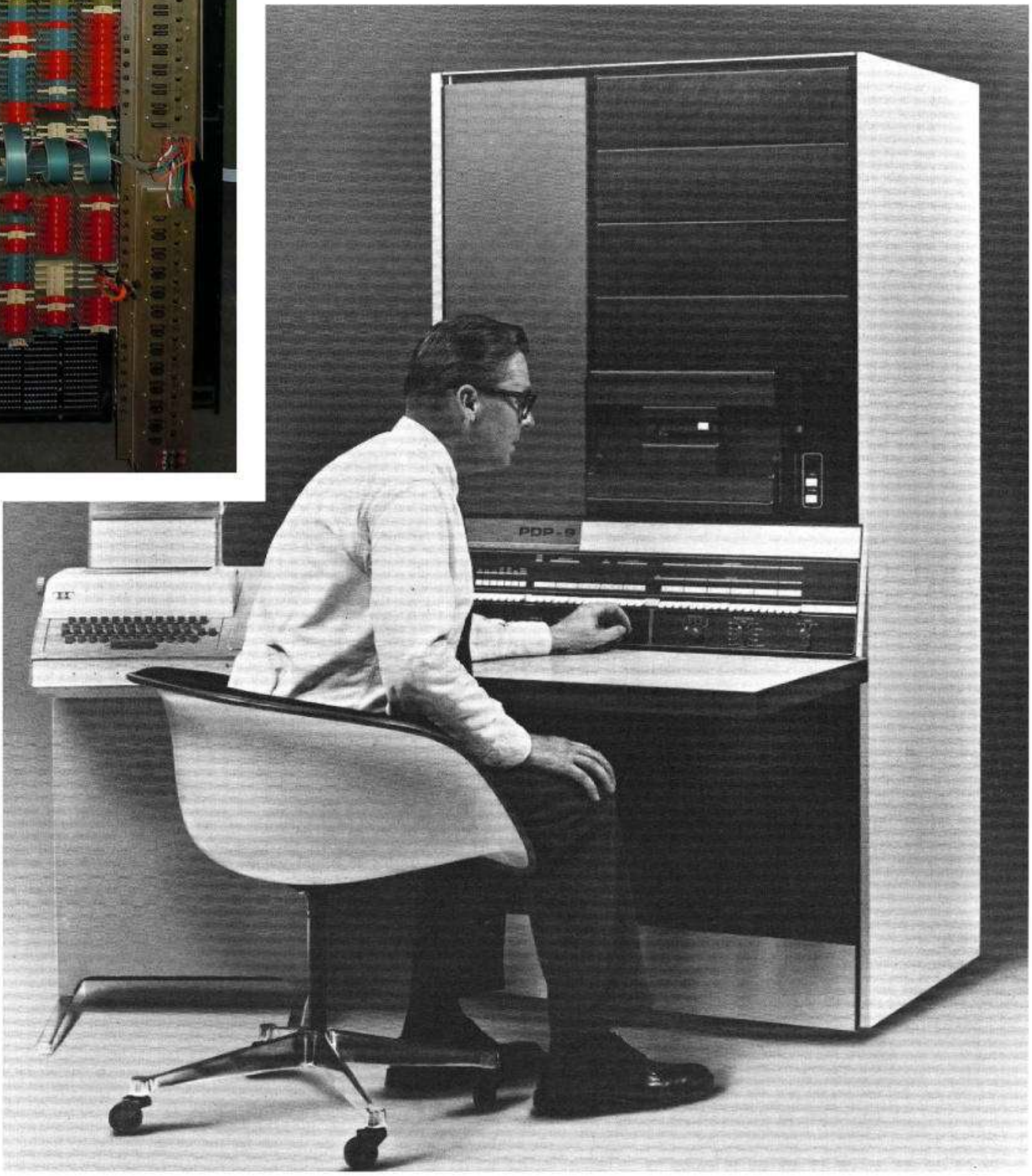
Warning! Frazzled Flip-Chips

Very old computers that use Flip-Chips have components operating at differing voltages, so there's a high chance of shorting them. You need a level shifter to convert and step-down voltages for safe operation.

magpi.cc/levelshifter



> Testing early transistor-only Flip-Chips is incredibly complicated because the voltages are all negative, and the Flip-Chips must be tested with varying input voltages and different loads



▲ Inside one of RICM's PDP-9 computers, showing the quantity of diodes and transistors involved

▶ This photo from the user manual shows just how huge the PDP-9 could get

Obstacles, of course

The Flip-Chip tester features 15 level-shifter boards. These step down the voltage so components with different power outputs and draws can operate alongside each other safely and without anything getting frazzled. Anders points out the disparity between the Flip-Chips' 0 and -3 V logic voltage levels and the +10 and -15V used as supply voltages. Huge efforts went into this level conversion to make it reliable and failsafe. Anders wrote the testing software himself and built the hardware "from scratch" using parts from Mouser and custom-designed circuit boards. The project took around two years, and cost around $500, of which the relays were a major part.

Anders favours Raspberry Pi because "it offers a complete OS, file system and networking in a neat and well packaged way" and says it is "a very good software platform that you really just have to do minor tweaks on to get right". He's run the tester on Raspberry Pi 3B, 4 and 5. He says it should also run on Raspberry Pi Zero as well "but having Ethernet and the extra CPU power makes life easier".

Although this is a fairly niche project for committed computer restorers, Anders believes his Flip-Chip tester can be built by anyone who can solder fairly small SMD components. Documenting the project so others can build it was quite a task, so it was quite helpful when Mike got in touch and was able to assist with the write-up. As a fellow computer restorer, Mike says the tester means getting RICM's PDP-9 working again "won't be such an overwhelming task. With the tester we can test and repair each of the boards instead of trying to diagnose a very broken computer as a whole."

▼ Anders previously used Raspberry Pi to recreate an old PDP-8 computer

Obsolescence Guaranteed

PiDP-8/I: RECREATING THE PDP-8/I

CELEBRATING 50 YEARS OF PROGRAMMED DATA PROCESSING

Introduction

The PiDP-8/I is a modern replica of the 1968 PDP-8/I computer. So far, more than 3,000 have been built - coming close to the production numbers for the real PDP-8/I (about 3800...)

It's open-source hardware, so schematics, design files & software are available here. As long as my stock of custom 'replica' switches lasts, I am also producing kits. There are large volume discounts on some of the other parts, so group ordering parts for a kit makes sense.

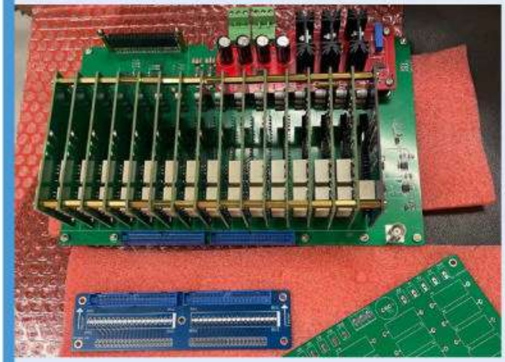
The sections to the left describe the PiDP in more detail. The web already contains lots of PDP-8 information, so these pages focus on the practical PiDP aspects: how to build & operate a PiDP-8 and importantly, what on earth to do with it...

Pictures of the current version

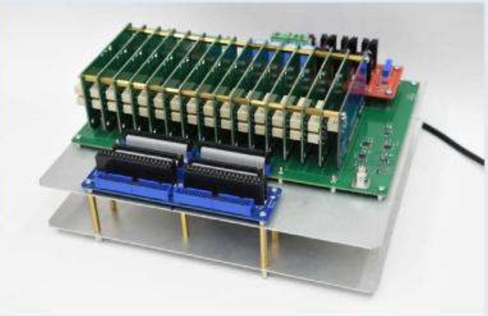
The PiDP-8, PiDP-11 & PiDP-10

Recommended: Gigatron
a computer w/o a microprocessor

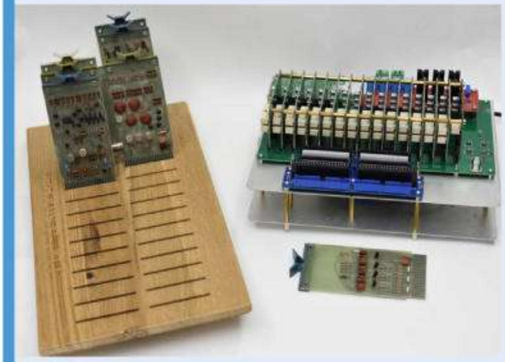
Relay reliable



01 Level shifter modules safely adjust voltages up and down so they operate correctly. The Flip-Chip modules use 0 and -3V volts for the logical 0 and 1. The tester can detect both logical and electrical failures.



02 Fifteen boards each take two pins and can act as an input/output level-shifter to 3.3 V TTL levels from 0 and -3 V levels.



03 When you power on Raspberry Pi, the brs-tester software runs from a GPIO subsystem shell. It takes different input parameters. Run `brs-tester --help` to get help text. To test a board, use a file with test vectors. Full instructions at **magpi.cc/brstester**.

Pixie clock

It's not a Nixie clock, it's a Pixie clock. By **Louis Wood**

MAKER

Louis Wood

Our Maker In Residence intern, Louis, has spent the summer making all sorts of cool Raspberry Pi builds.

magpi.cc/louiswood

This project was inspired by Nixie tubes, an old technology used for displaying digits. The trouble with Nixie tubes is that they are quite dangerous, requiring 170 volts to produce their characteristic glow, and very expensive, since they are no longer mass produced. A safer, more modern approach is using LEDs to light the edges of engraved acrylic.

Electronics

The electrical design of this project is simple and uses a single 300-LED-long strip of NeoPixels, a Raspberry Pi 4, and a 5V power supply. To provide ample current to the Raspberry Pi and the pixels, I went with an 8A supply.

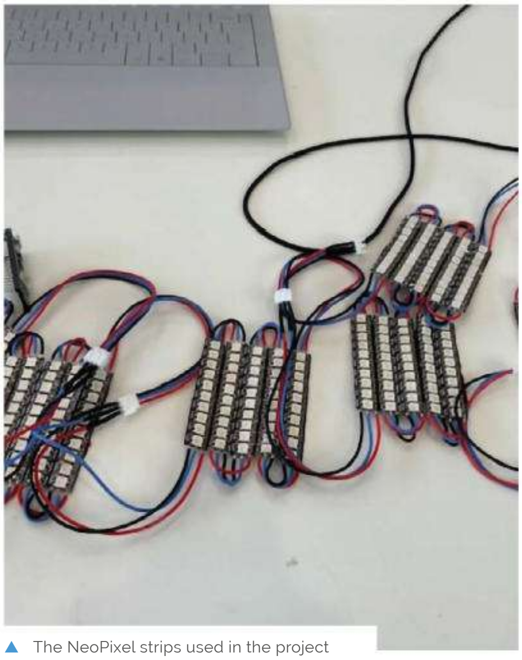
While there are only a few electrical components a lot of soldering was required. The LED strip needed to be cut into 30 segments, which I then soldered back together with short wires to allow the strip to snake. Between sets of strips, I soldered longer wires with crimped connectors for easy linking of tubes.

Tubes, base, and programming

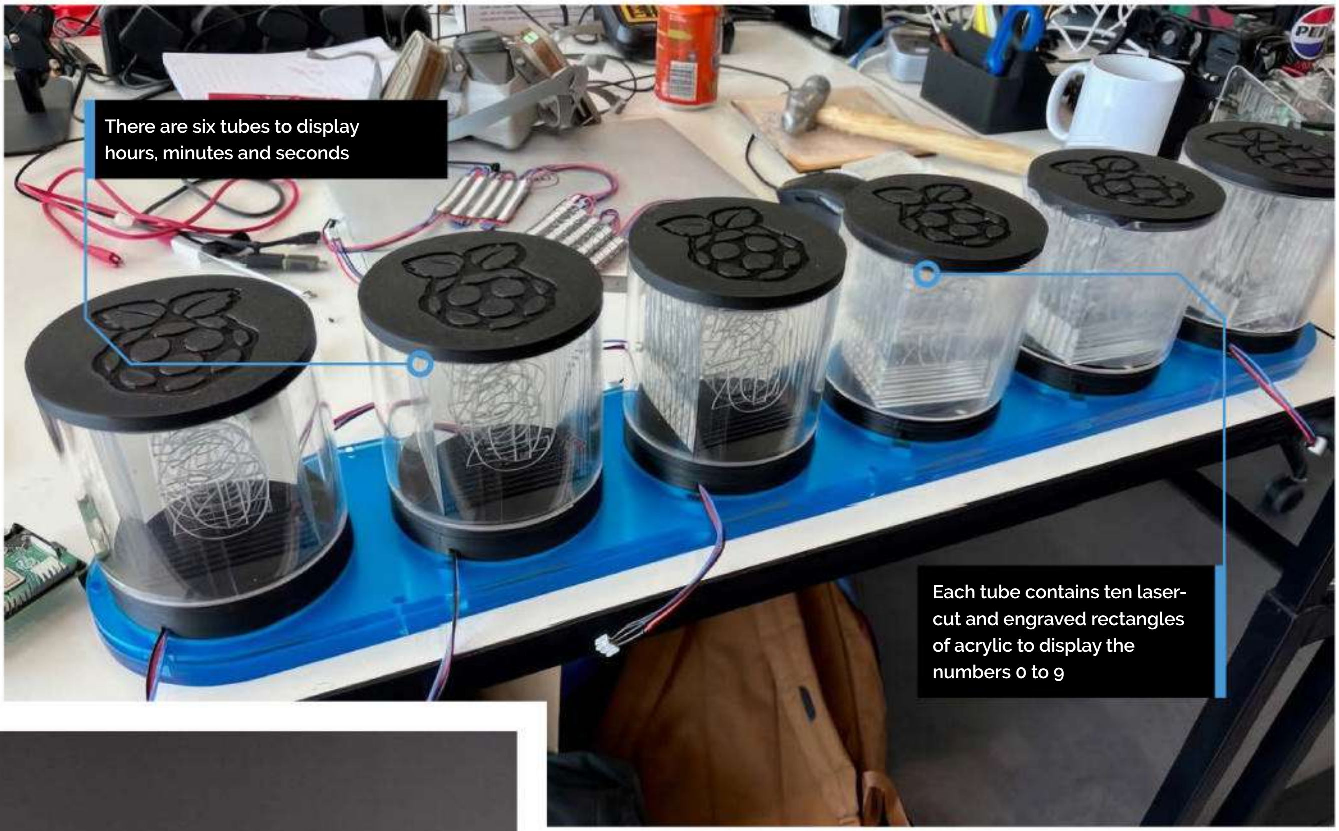
Each tube needs to display digits from 0 to 9, so each contains ten laser-cut and engraved rectangles of acrylic. To provide more of a tubular appearance, I surrounded the digits with a transparent cylindrical enclosure. All this acrylic is housed between two 3D-printed discs. The LEDs sit below this assembly in their own 3D-printed enclosure, and the two parts come together with recessed magnets for a clean look.

Since I was building a clock to display hours, minutes and seconds, six tubes were required. To hold these in position, together with the Raspberry Pi, I laser-cut a base from more acrylic, with 3D-printed standoffs between the top and bottom.

I loaded the Raspberry Pi with a simple program, using `datetime` to fetch time information and then display it on the clock by lighting up the appropriate LEDs. Each digit has five LEDs associated with it, so I created arrays to make it easy to address the pixels for a particular digit.

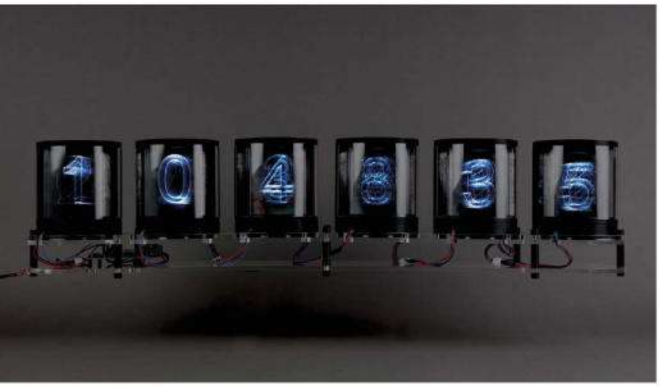


▲ The NeoPixel strips used in the project



There are six tubes to display hours, minutes and seconds

Each tube contains ten laser-cut and engraved rectangles of acrylic to display the numbers 0 to 9



Pixie Clock displaying LED numbers

> The clock is currently doing its regular job in the Maker Lab at Pi Towers

Laser-cutting acrylic

For example, one of these arrays looks like this:

```
h1 = [
    [0, 19, 20, 39, 40],
    [1, 18, 21, 38, 41],
    [2, 17, 22, 37, 42],
    [3, 16, 23, 36, 43],
    [4, 15, 24, 35, 44],
    [5, 14, 25, 34, 45],
    [6, 13, 26, 33, 46],
    [7, 12, 27, 32, 47],
    [8, 11, 28, 31, 48],
    [9, 10, 29, 30, 49]
]
```

`h1` means the array is for the first of the two hour digits. To light up the digit 7 for the first hour digit, you would turn on all the pixels in the list `h1[7]` [*I don't want to think about why you might need to show that it's 70-something hundred hours – Ed*]. The slightly strange pixel order is due to the way the NeoPixels snake their way through the clock design.

The clock is currently doing its regular job in the Maker Lab at Pi Towers, but our Head of Social asked if I could tweak the code to turn it into a likes counter for our TikTok account (**magpi.cc/rpi-tiktok**), so once I've managed to do that, it'll have a whole new purpose.

Quick FACTS

- Nixie tubes were usually filled with neon…
- …which require a high voltage (around 170w)
- Pixie uses 300 NeoPixels instead
- The LED lights are a lot safer
- It's being adapted as a TikTok likes counter

Pi-CARD AI assistant

An offline take on a Raspberry Pi voice assistant makes canny use of AI without sharing user data, intriguing **Rosie Hattersley**

MAKER

Noah Kasmanoff

Data scientist Noah Kasmanoff is fascinated by technology to make people's lives easier and was drawn to Raspberry Pi due to its very accessible applications.

magpi.cc/picardgit

As an astronomy and physics student, Noah Kasmanoff ran Python simulations to help design scientific instruments. As a data scientist, Noah is interested in how technology makes our lives easier, and recently set about creating an AI assistant to help us spend less time on our phones.

The result, Pi-CARD, is far from the only Raspberry Pi AI assistant, but it's the first one we've encountered whose name pays tribute to the legend that is Jean-Luc Picard.

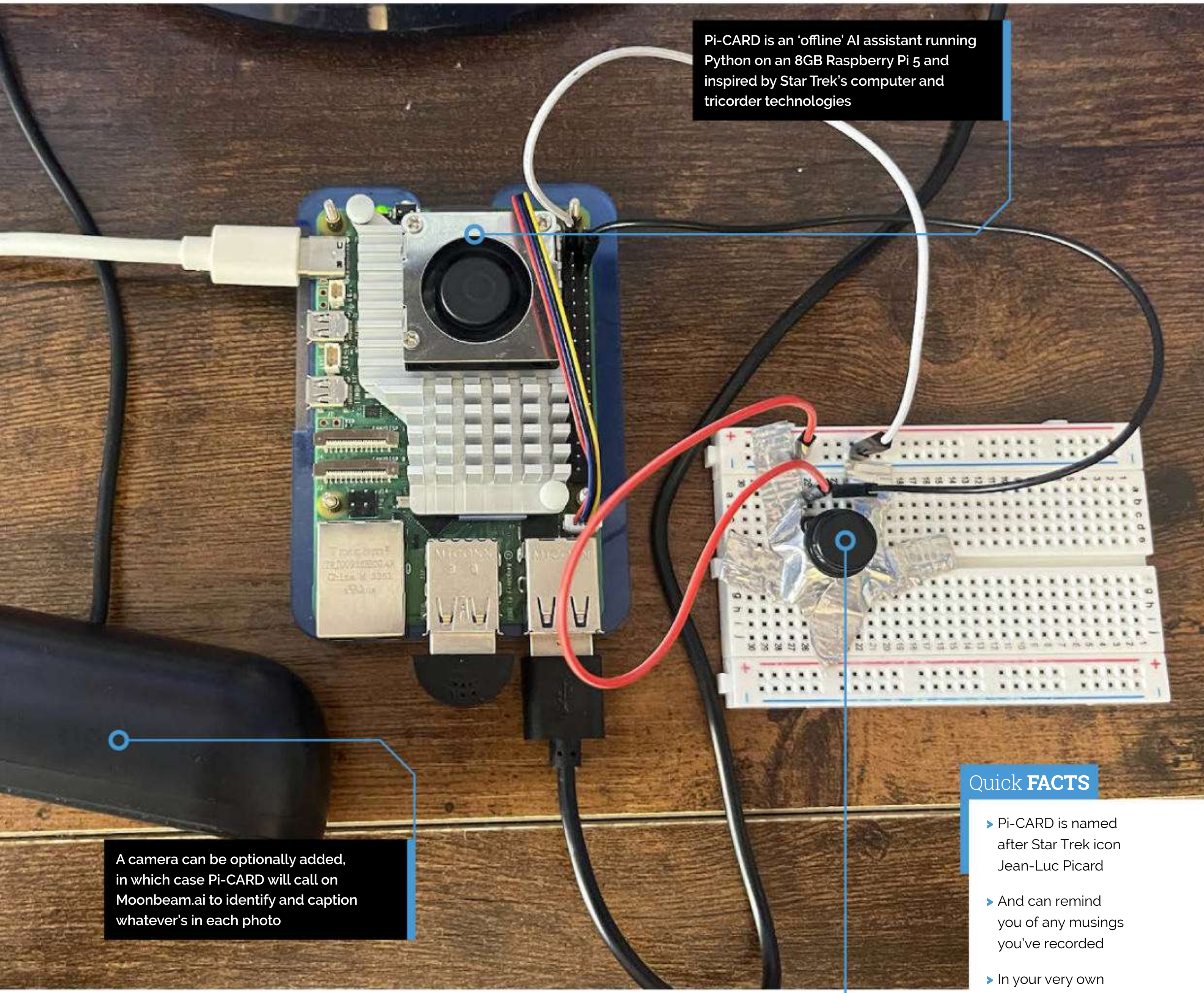
Noah's aim with Pi-CARD (Camera Audio Recognition Device) was to see how useful he could make an AI assistant that runs entirely on a single small device and with privacy and data protection built in. "The inspiration comes from *Star Trek*'s computer and tricorder technologies, essentially trying to maximise the trade-offs between a smart device while making it as small and cheap as possible."

Say what you see

After researching existing Raspberry Pi AI assistants, Noah decided to try using Ollama (**ollama.com**), a command line-based tool for downloading and running open-source LLMs (large language models). He was particularly intrigued to discover how effective an entirely offline AI assistant can be.

Pi-CARD can perform the sorts of tasks you'd expect from a standard LLMs such as ChatGPT: Llama.cpp and Whisper.cpp are used for vision and automatic speech recognition respectively. However, unlike Alexa and other voice assistants: "once your wake word has been said, you are officially in a conversation and do not need to constantly repeat the wake word. Pi-CARD then listens for commands until you say something like 'stop', 'exit' or 'goodbye'."

Noah adds that if you attach a camera, you can also ask Pi-CARD to take a photo, describe what it sees, and then details about that image. For an idea of what to expect, upload a photo to **moondream.ai**, the image recognition tool used by Pi-CARD.



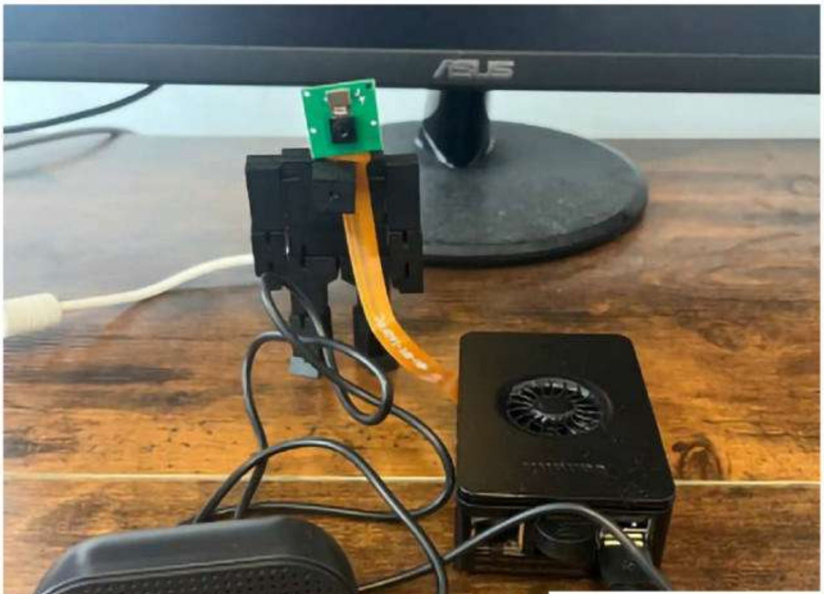
Pi-CARD is an 'offline' AI assistant running Python on an 8GB Raspberry Pi 5 and inspired by Star Trek's computer and tricorder technologies

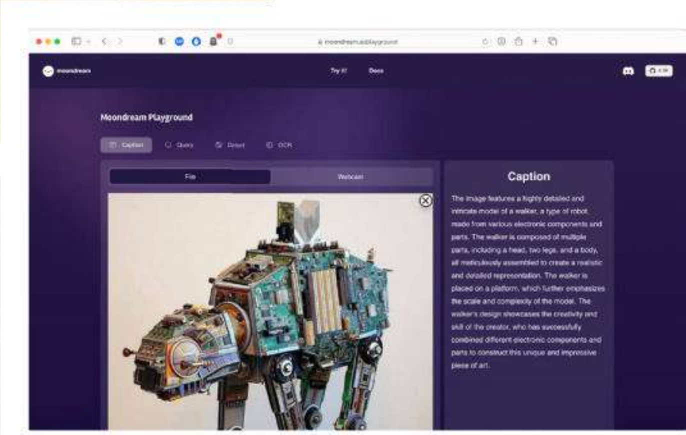
A camera can be optionally added, in which case Pi-CARD will call on Moonbeam.ai to identify and caption whatever's in each photo

Raspberry Pi's GPIO ports meant maker Noah could add a button to wake it. Pi-CARD uses Ollama LLM to serve up notes and other information stored locally

Quick FACTS

- Pi-CARD is named after Star Trek icon Jean-Luc Picard
- And can remind you of any musings you've recorded
- In your very own captain's log
- Noah is working on a portable version of Pi-CARD
- That he can call up whenever he's out on a run

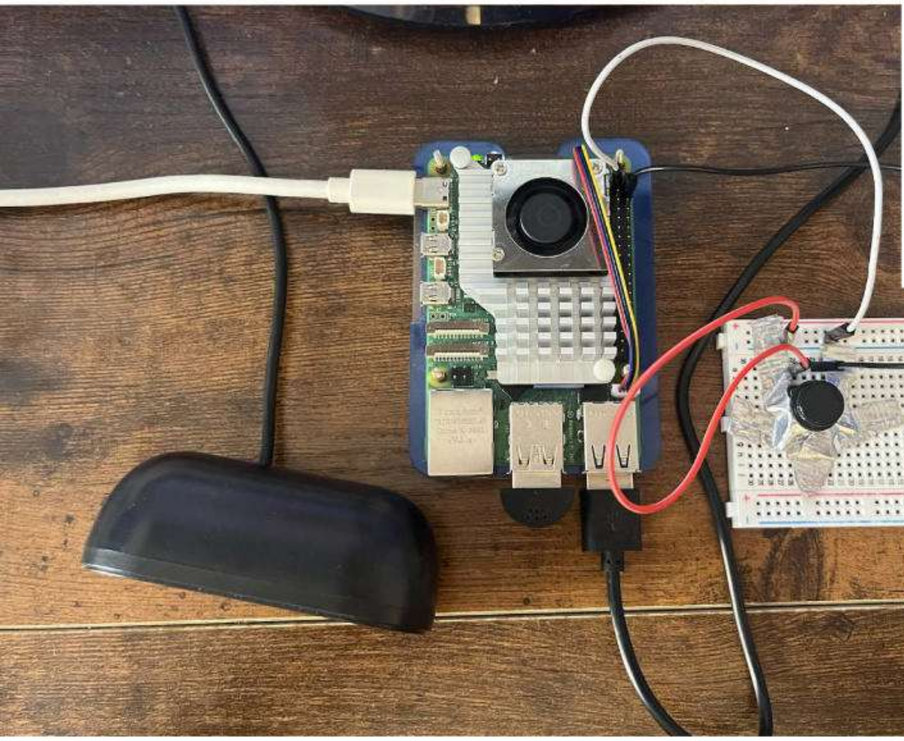




Moondream Playground

Caption

The image features a highly detailed and intricate model of a walker, a type of robot, made from various electronic components and parts. The walker is composed of multiple parts, including a head, two legs, and a body, all meticulously assembled to create a realistic and detailed representation. The walker is placed on a platform, which further emphasizes the scale and complexity of the model. The walker's design showcases the creativity and skill of the creator, who has successfully combined different electronic components and parts to construct this unique and impressive piece of art.



◀ Add a webcam or a Raspberry Pi HQ camera module and Pi-CARD will describe photos

▲ Upload a photo to try out **moondream.ai** photo recognition for yourself

◀ Raspberry Pi's GPIO ports improve flexibility, including adding buttons

Co-ordinated approach

His biggest challenge was co-ordinating the whole system and reducing the operating latency as much as possible. "To go from recording a message, transcribing it, passing it through an LLM, and dictating that response out loud is a sophisticated process", one that needs to happen within a second or two. Otherwise, Noah realised, there would be "a huge hit on the overall feel of Pi-CARD".

Noah is using an 8GB Raspberry Pi 5, and says this is "the right hardware", although it could also run on other platforms that support Python. He is working on a Docker implementation to make it easier to run." The project cost around $200 (£155) to build, with most parts bought online and code written by Noah.

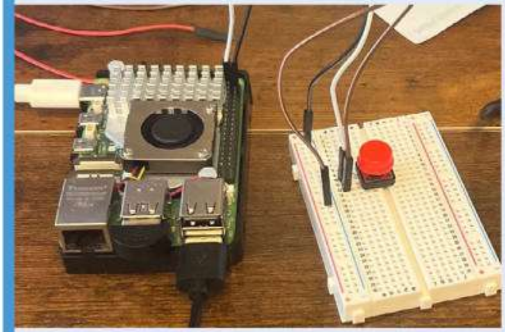
He shared his still "ongoing and evolving" Raspberry Pi 5 project on GitHub earlier this year to some acclaim. "It got a lot more attention than I thought it would." He says it currently looks a bit like Google Home or Alexa "but I hope to push this a lot further".

> " An AI assistant to help us spend less time on our phones "

Would-be makers will need a Raspberry Pi, a speaker, a microphone, and a optional camera, with code and instructions detailed on Noah's GitHub (**magpi.cc/picardgit**). Pi-CARD could also be connected to a network and server if you prefer to use cloud AI services and don't mind sharing your user data. Costs could be minimised with a cheaper case and different SBC, but Noah feels Raspberry Pi's solid community and build quality are easily worth it. "Any of the problems I had with the device specifically could be solved by looking things up." And although he didn't realise it immediately, Noah says the biggest advantage to using Raspberry Pi 5 was having GPIO access. "This meant I could attach a button to the whole thing, which changed how it worked for the better."

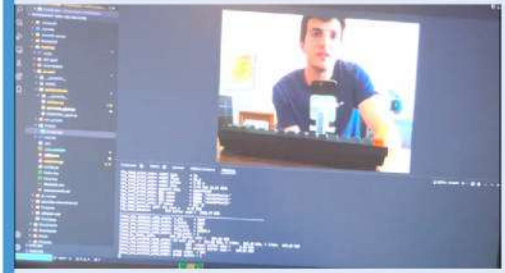
His plan is to make a portable version he could potentially take out and about while running. Since Pi-CARD runs locally, rather than drawing on cloud-based resources, its speed is limited, but Noah is also working hard to reduce its latency and make the responses more efficient.

Spy-free AI



01 Pi-CARD will run on almost any version of Raspberry Pi and processes everything locally rather than sharing user data with a cloud-based AI server. Full setup instructions are at **magpi.cc/picardgit**.

02 Unlike some AI assistants, Pi-CARD does not listen out constantly for a wake word. To initiate it, run the wake script, followed by the shutdown script when you've finished.



03 Noah's YouTube demo (**magpi.cc/picardyt**) shows Pi-CARD using Moondream2 for image recognition. Images are limited to 378×378 pixels but the description is still pretty accurate.

Smart Pill Dispenser

Judges at this year's Coolest Projects challenge were hugely impressed by a pragmatic teenager's Raspberry Pi-based approach to age-related forgetfulness. By **Rosie Hattersley**

MAKER

Andrew Zhang

High school student Andrew Zhang has been programming since 6th grade, and has a passion for creating cool software and using hardware to bring it to life.

magpi.cc/smartpillvid

Every year, the Raspberry Pi Foundation runs and helps judge Coolest Projects, a competition in which young coders and makers from around the world are encouraged to use technology in innovative ways. Among the hundreds of entries to this year's Coolest Projects challenge, 15-year-old Andrew Zhang's Smart Pill Dispenser stood out for the considered approach to solving a problem he'd encountered first-hand – his grandmother sometimes forgets to take important medications, and Andrew's informal research among her peers indicated it was a fairly common scenario – as well as the way he went about designing a means of solving it, using Raspberry Pi Zero, and documenting everything in superb detail.

Aiming high

Aspiring software developer engineer Andrew is still a junior high-school student but, given the quality of his Smart Pill Dispenser project, it came as little surprise when he told us his technical chops have already been recognised. He also helps out with web development as part of a backend team for various nonprofits that have competed in (and won) seven hackathons.

▼ Andrew with his smart pill dispenser project, which won a Coolest Projects award

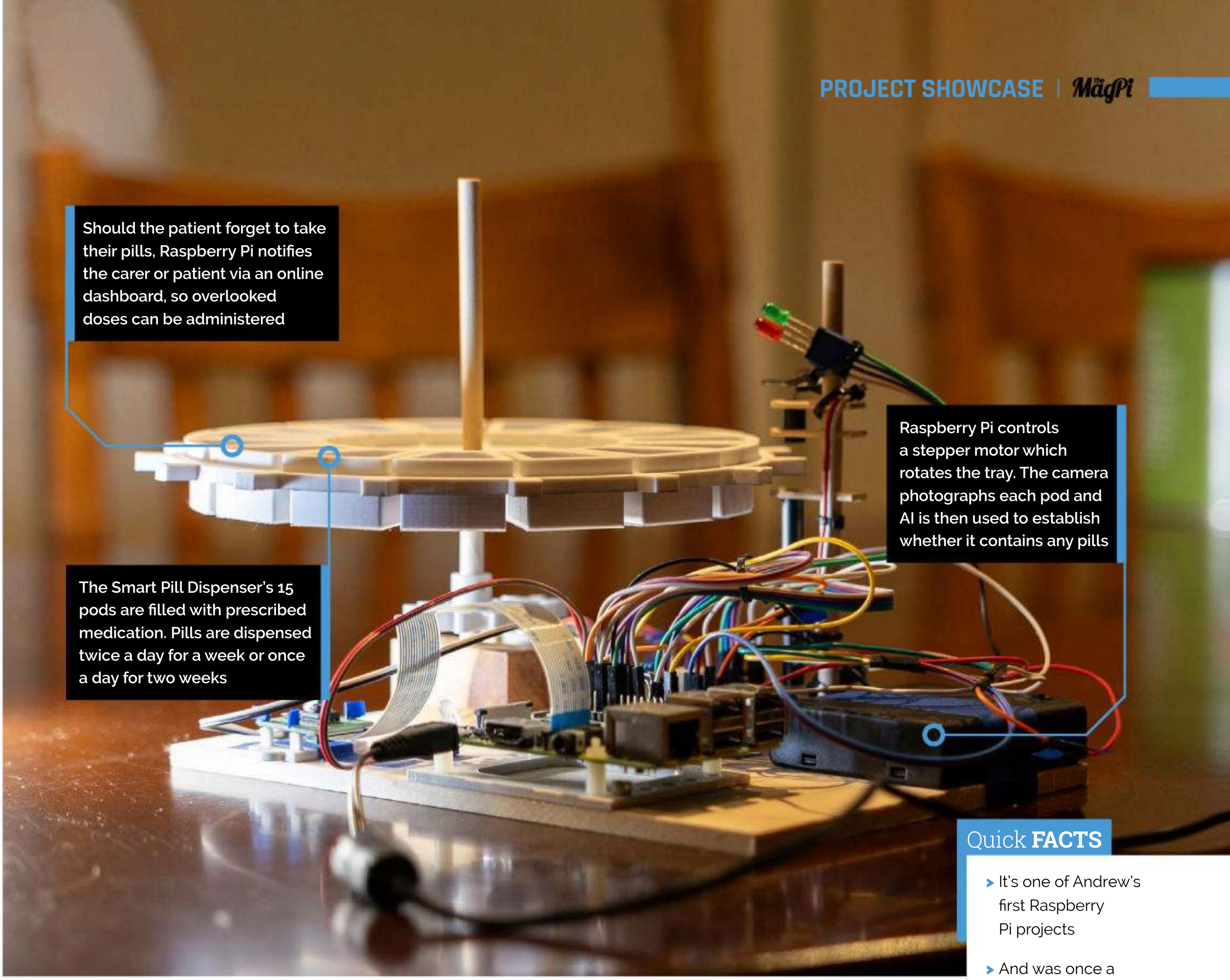
▲ Pills sorted into doses and stored in their smart pill dispenser pods

His Coolest Projects entry aimed to "leverage AI and internet technology to solve the problem in a way that is reliable, convenient, and affordable. Keen photographer Andrew realised that training a camera on a pill box to check whether it was still empty long after medication time could provide an invaluable failsafe for patients and their carers.

Andrew's motivation for the Smart Pill Dispenser stemmed from his grandmother's "occasional forgetfulness" around conventional medication containers. He designed a prototype smart dispenser that can automatically verify whether a patient has taken their medication, thus helping his grandmother and her elderly friends. Carers can monitor remotely and provide prompt intervention when necessary.

Ongoing refinements

The smart pill dispenser consists of Raspberry Pi 3B, stepper motor and driver board, a Raspberry Pi Camera Module and a 3D-printed dispensing tray on a spindle. The prototype version has a cardboard base and scrap wood supporting tray. However, a 3D-printed baseboard to replace these is in the works. The stepper motor turns the pill container tray so the camera can take pictures through each container's clear bottom. A photoresistor and several LEDs form a photogate which acts as a



Should the patient forget to take their pills, Raspberry Pi notifies the carer or patient via an online dashboard, so overlooked doses can be administered

The Smart Pill Dispenser's 15 pods are filled with prescribed medication. Pills are dispensed twice a day for a week or once a day for two weeks

Raspberry Pi controls a stepper motor which rotates the tray. The camera photographs each pod and AI is then used to establish whether it contains any pills

Quick FACTS

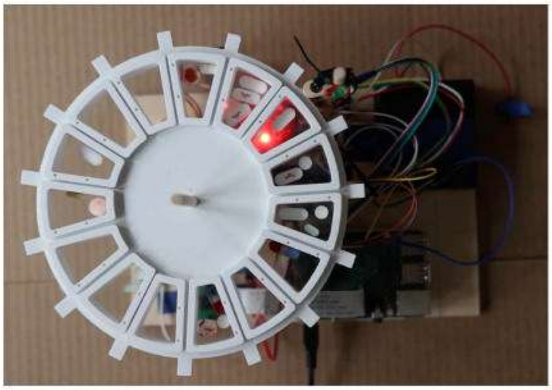
- It's one of Andrew's first Raspberry Pi projects
- And was once a sweetie dispenser…
- …that wouldn't pay out if you were greedy
- Andrew recently made a Raspberry Pi-powered FM radio with built-in AI
- And an AI-powered personal FM radio station too

fill light and shows the device's status. Four AA batteries power the motor, but Andrew is looking into ways to power both this and Raspberry Pi from a single DC-DC power supply. He also wants to replace the existing photogate with a better-trained AI to create a more compact unit.

Raspberry Pi was the ideal choice for this project, says Andrew. Its GPIO interface allows communication with essential hardware components such as the stepper motor, LEDs and photogate as well the Raspberry Pi camera. Raspberry Pi is also used to run TensorFlow Lite and the Django framework. "Its small size and low power consumption are also essential for making this project practical. Most importantly, the vast amount of widely available online tutorials from the Raspberry Pi maker community significantly shortened my project development time and helped to make it a reality", said Andrew. The system cost $90 to build and $4 a year to run and proved reliable, accurate, cost-effective and easy to use. In fact, Andrew believes, "with improvements, it has the potential to be a widely used medical device serving large groups of patients".

Andrew explains his project in methodical detail on Vimeo: **magpi.cc/smartpillvid**.

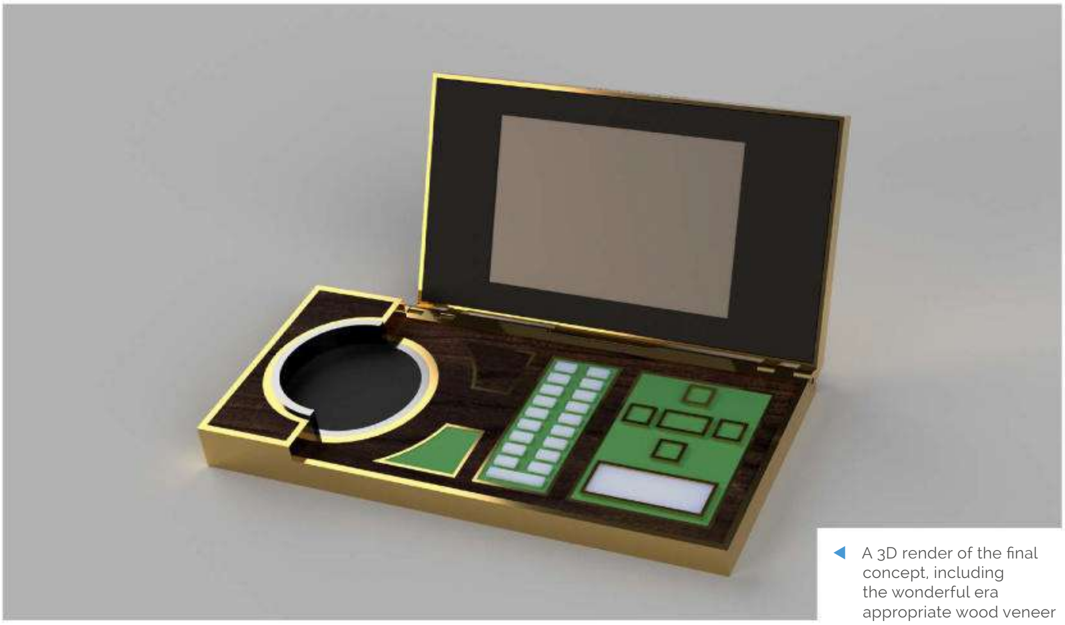
> Andrew's motivation for the Smart Pill Dispenser stemmed from his grandmother's occasional forgetfulness around conventional medication containers



A Raspberry Pi Camera Module photographs each pill pod to check its contents

TVA TemPad

Recreating a retro-futuristic Marvel prop can be done with just an RP2040. **Rob Zwetsloot** observes the timeline it appears in



A 3D render of the final concept, including the wonderful era appropriate wood veneer

MAKER

Matt Gray

A YouTuber of over 15 years, Matt makes amazing things using tech, including a hovercraft pub, 3D-printed human cells, and pyrotechnics.

magpi.cc/tvatempad

Protecting the timelines of reality, the Time Variance Authority (TVA) is an anachronistic organisation in the Marvel universe that resembles a drab 1970s office full of retro-futuristic technology that allows its agents to travel to any time. This includes the TemPad, a tablet that shows information about timelines and allows you to travel via temporal doors – it's a neat little prop, and it caught maker and YouTuber Matt Gray's attention.

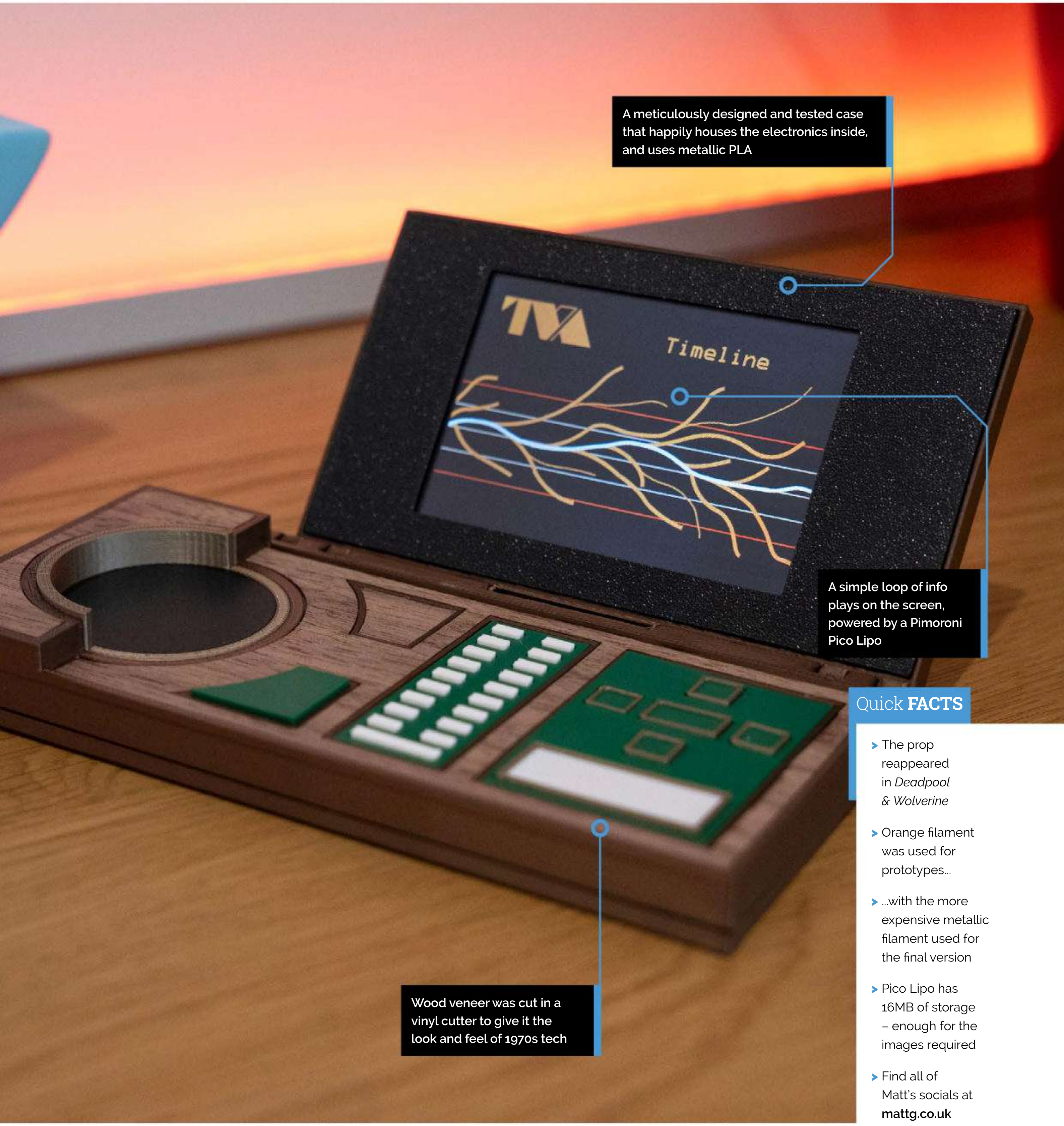
"Watching the *Loki* TV series, I spotted the little handheld folding computer thing that the Time Variance Authority use and liked the look of it," Matt says. "After a quick search, the only attempt I could see of someone making one was just a non-functional 3D print. It looked OK but it didn't do anything, and it just had a bit of plastic where the screen should be."

Clearly, it was a sign for Matt to make his own, pushing his own knowledge and skill to the limits in the process.

Time and again

Matt spent about a month making the TemPad – iterating on each part until he was happy with it. "I used parts from Pimoroni and Adafruit because they both make really straightforward and well-documented things that take so much of the friction away from prototyping something," Matt explains. "Once I'd put together a proof-of-concept, I used Fusion 360 to create a 3D model that not only looks like a TemPad, but could also fit all the electronics in. This has been by far the most complicated thing I've ever designed, so it took days! I used metallic PLA 3D printing filament to make the body, but to fit in with its retro-futuristic aesthetic the prop in Loki looks like it has wooden parts. So I put some real wood in a vinyl cutter to make walnut veneer layer to give it that extra touch of realism."

The heart of the project is a Pimoroni Pico Lipo (**magpi.cc/picolipo**), a Pico-like board that can connect to a LiPo battery to improve portability and cut down on battery space.



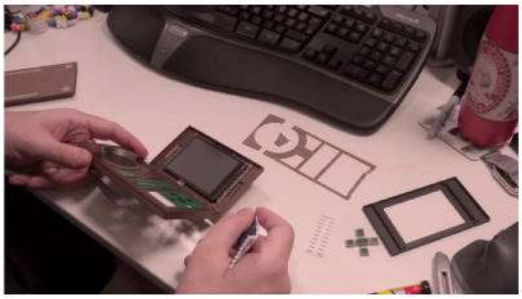
A meticulously designed and tested case that happily houses the electronics inside, and uses metallic PLA

A simple loop of info plays on the screen, powered by a Pimoroni Pico Lipo

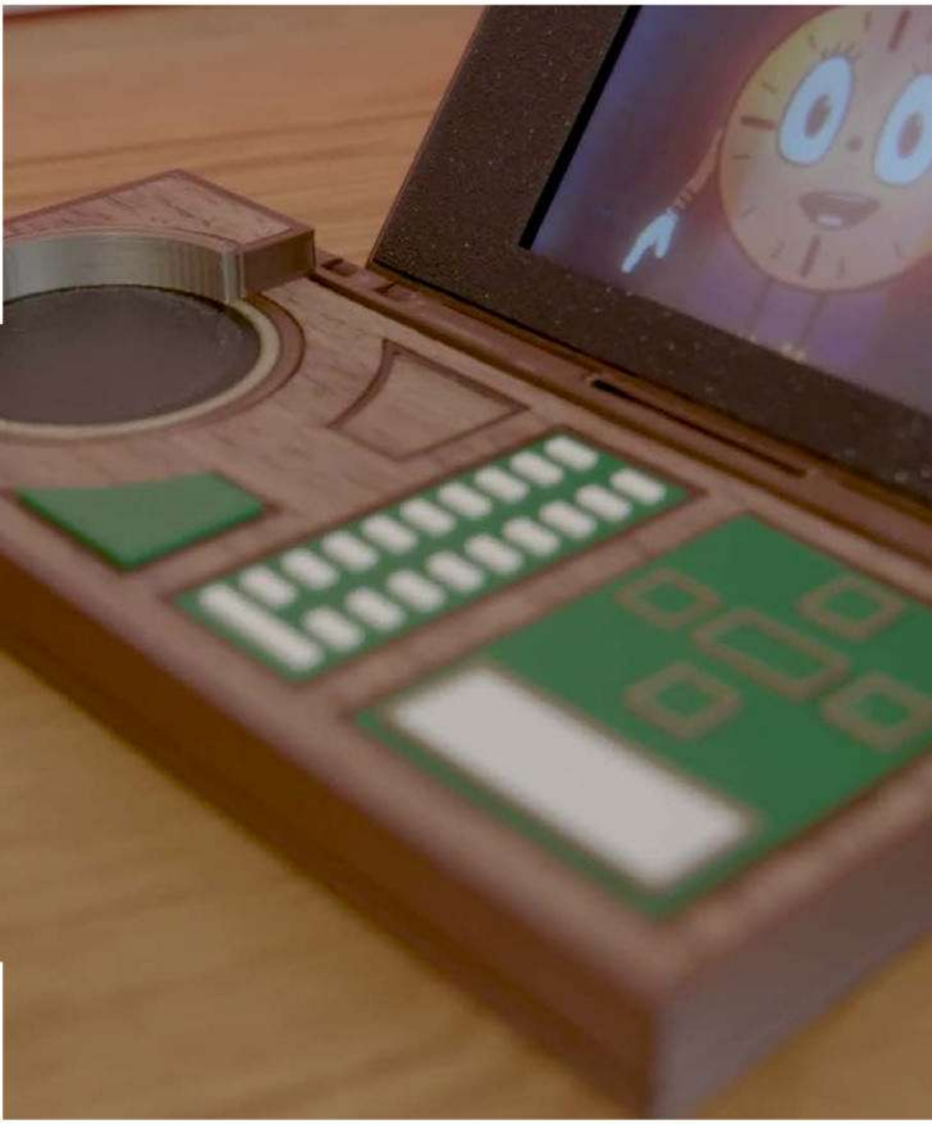
Wood veneer was cut in a vinyl cutter to give it the look and feel of 1970s tech

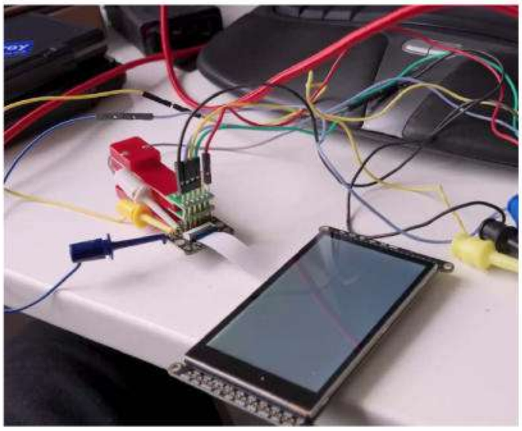
Quick FACTS

- The prop reappeared in *Deadpool & Wolverine*
- Orange filament was used for prototypes…
- …with the more expensive metallic filament used for the final version
- Pico Lipo has 16MB of storage – enough for the images required
- Find all of Matt's socials at **mattg.co.uk**

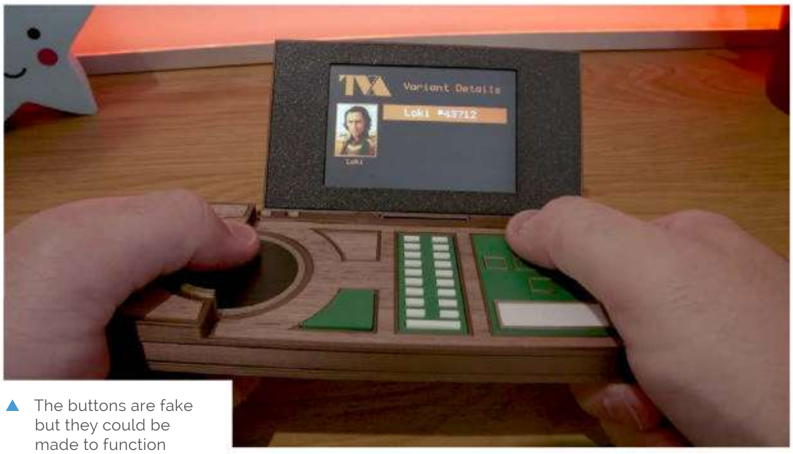


The parts are stuck together with superglue – a mishap with which meant reprinting a part

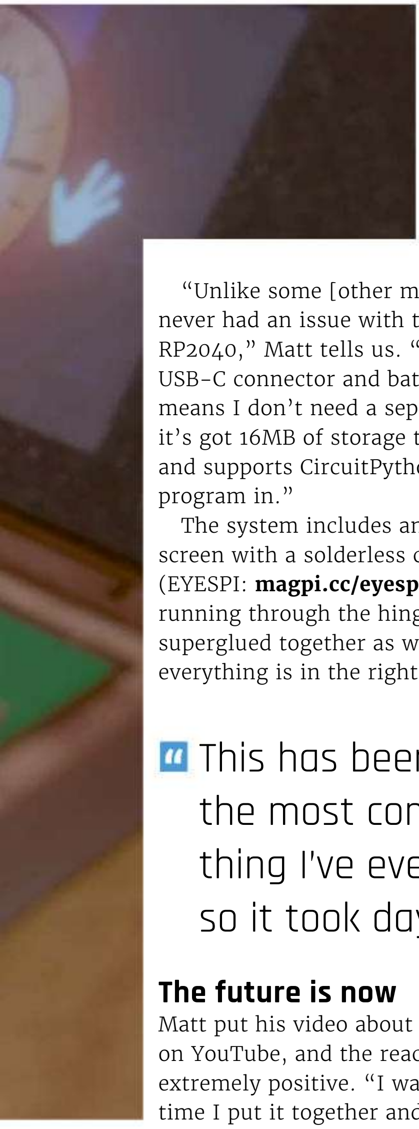




A final test with all the parts being used in the final product



The buttons are fake but they could be made to function



The wood veneer gives it the movie prop feel Matt was going for

"Unlike some [other microcontrollers], I've never had an issue with the capability of the RP2040," Matt tells us. "And the Pico Lipo's USB-C connector and battery management means I don't need a separate board or cables, it's got 16MB of storage to fit my images on, and supports CircuitPython which I prefer to program in."

The system includes an Adafruit 3.5" TFT screen with a solderless connector system (EYESPI: **magpi.cc/eyespi**), with a cable running through the hinge. The device is superglued together as well, so making sure everything is in the right place first is a must.

> This has been by far the most complicated thing I've ever designed, so it took days

The future is now

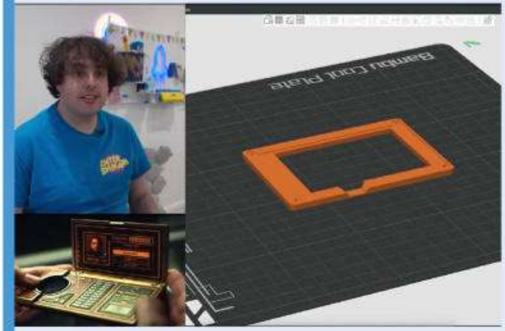
Matt put his video about the TemPad up on YouTube, and the reactions have been extremely positive. "I was so giddy the first time I put it together and held it," Matt says. "Everyone who's seen it in person has been wide-eyed too!"

And despite how happy he is with the build, Matt has some ideas on how to further improve it: "Currently the buttons are all fake, but I've made it in such a way that they could be functional in the future," Matt reveals. "At the time, that felt like scope creep, but if I end up showing this off in person more, then working buttons would be a great upgrade, and I'll probably add a little weight in the base to make it feel a little less top-heavy!"

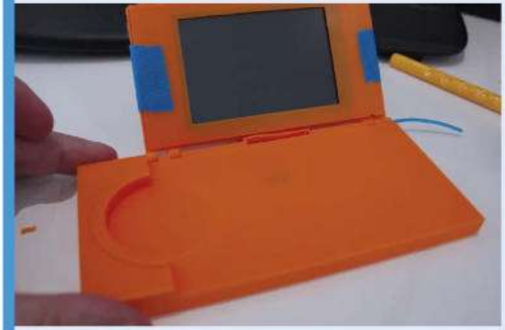
You can check out Matt's YouTube (**magpi.cc/mattgray**) channel for the Tempad and other Raspberry Pi projects – and maybe he'll find this universe's anchor being in a future video? "Curiously my research suggests the best place to look may be Xandar," Matt admits. "Or Scunthorpe."

Iterative design

01 The basics of the electronics were tested to make sure they all worked as expected before designing the case around them.



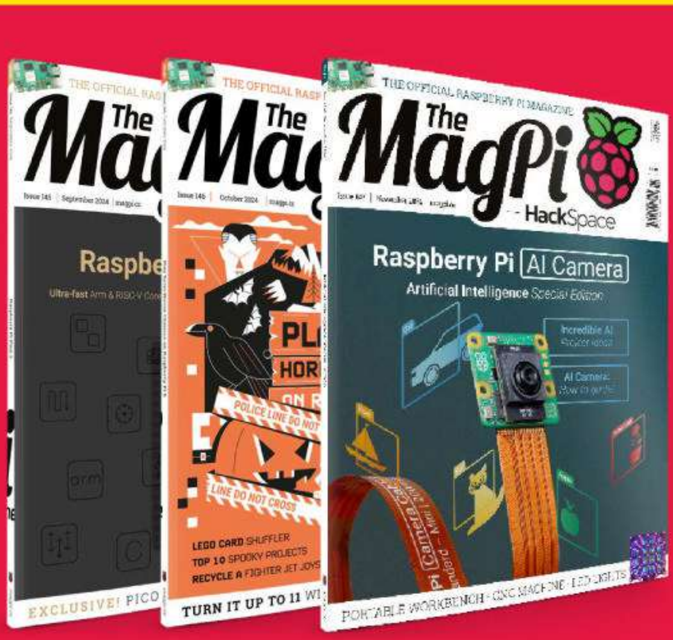
02 Using Fusion 360, Matt printed off a few test cases until he was satisfied with his design and tolerances – everything is very tight in the case!



03 The final proof of concept before starting to print the real cases for the final product – with filament used as a hinge pin!

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W


The MagPi
Raspberry Pi AI Camera
Artificial Intelligence Special Edition
HackSpace
LEGO CARD SHUFFLER
TOP 10 SPOOKY PROJECTS
RECYCLE A FIGHTER JET JOYSTICK
TURN IT UP TO 11
EXCLUSIVE! PICO
PORTABLE WORKBENCH · CNC MACHINE · LED LIGHTS

+

Subscriber benefits

- FREE Delivery
  Get it fast and for FREE
- Exclusive Offers
  Great gifts, offers, and discounts
- Great Savings
  Save up to 23% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- £10 (UK only)

Subscribe for 6 Months

- Free Pico W
- 6 issues of The MagPi
- £30 (UK) $43 (USA)
  €43 (EU) £45 (Rest of World)

☎ Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a six-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

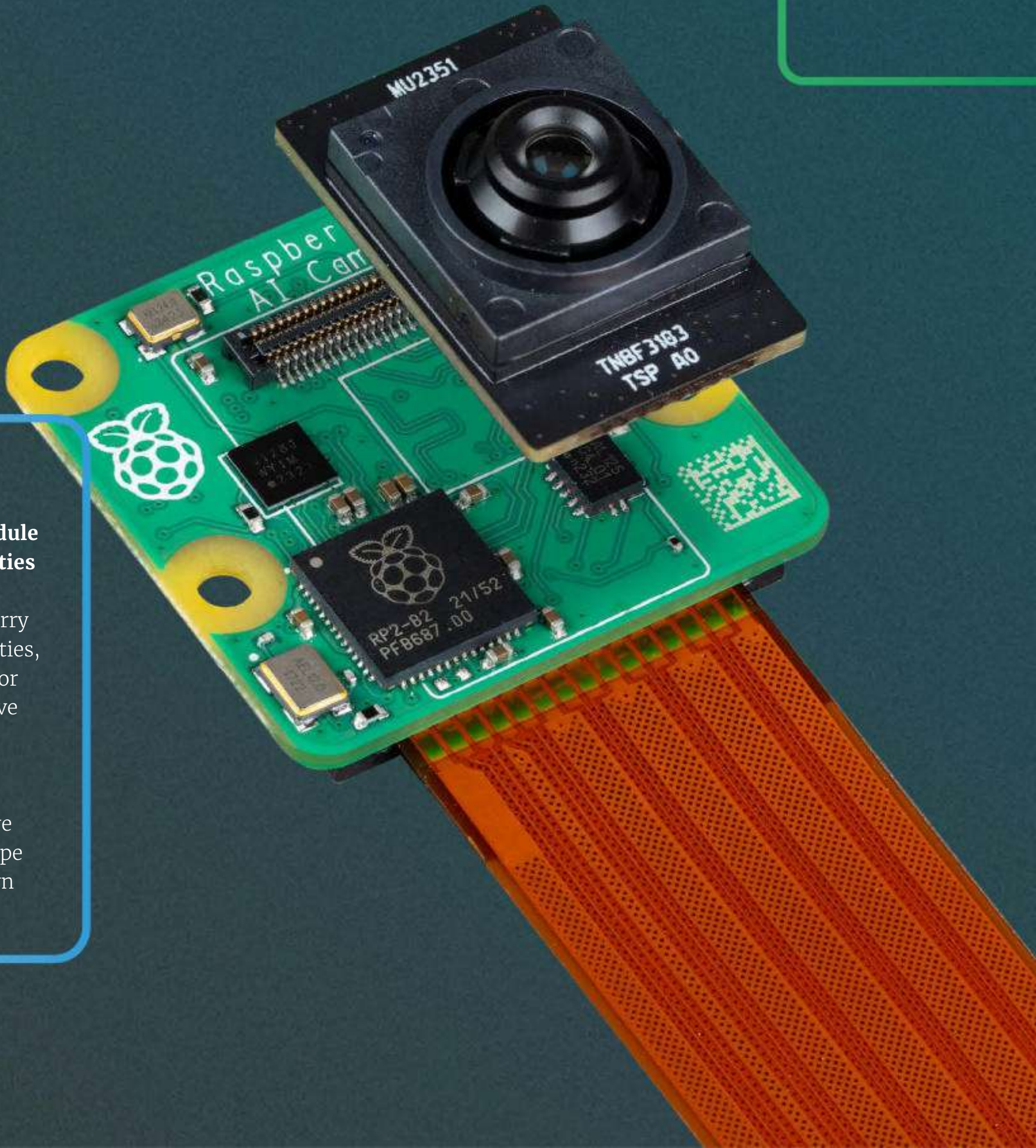
From **£2.29**

Available on the App Store

GET IT ON Google Play

AI projects

Add artificial intelligence to your Raspberry Pi designs. By **Phil King**



With their powerful AI accelerator modules, Raspberry Pi's Camera Module and AI Kit open up exciting possibilities in computer vision and machine learning. The versatility of the Raspberry Pi platform, combined with AI capabilities, opens up a world of new possibilities for innovative smart projects. From creative experiments to practical applications like smart pill dispensers, makers are harnessing the kit's potential to push the boundaries of AI. In this feature, we explore some standout projects and hope they inspire you to embark on your own AI-driven adventure.

AI Kit projects

Discover innovative ideas powered by the Raspberry Pi AI Kit

Peeper Pam

VEEB

magpi.cc/peeperpam

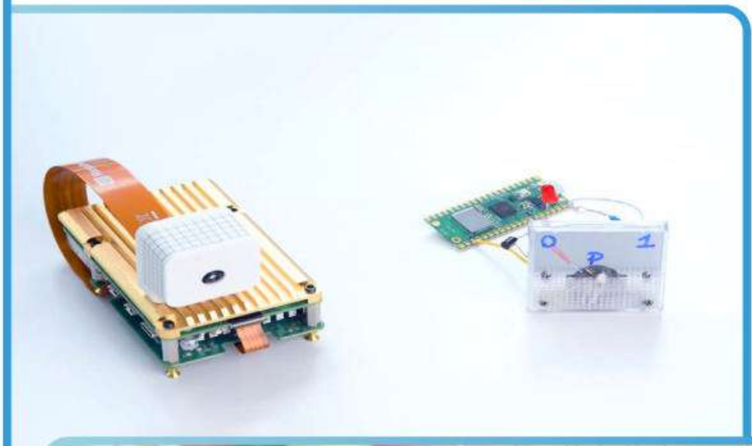
AI computer vision can identify objects within a live camera view. In this project, VEEB's Martin Spendiff and Vanessa Bradley have used it to detect humans in the frame, so you can tell if your boss is approaching behind you as you sit at your desk!

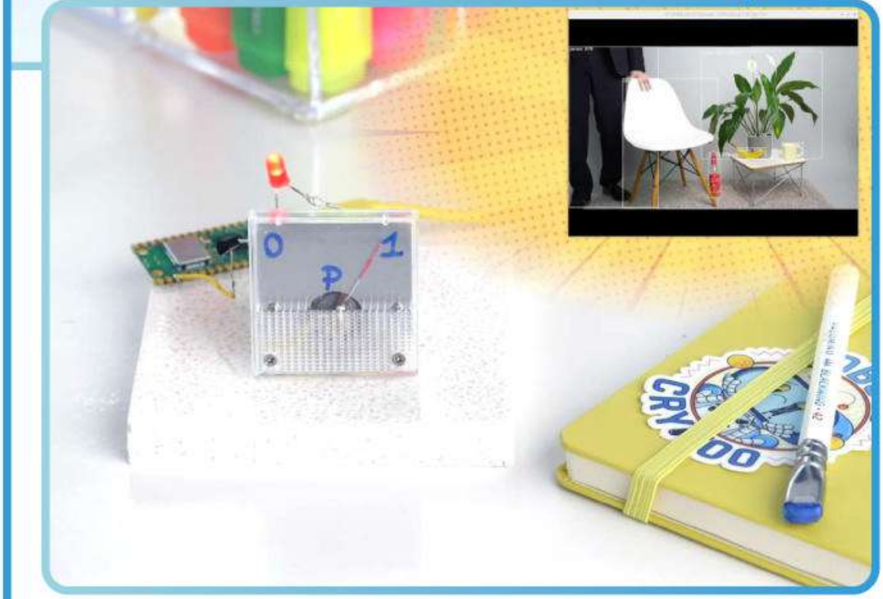
The project comprises two parts. A Raspberry Pi 5 equipped with a Camera Module and AI Kit handles the image recognition and also acts as a web server. This uses web sockets to send messages wirelessly to the 'detector' part – a Raspberry Pi Pico W and a voltmeter whose needle moves to indicate the level of AI certainty for the ID.

Having got their hands on an AI Kit – "a nice intro into computer vision" – it took the pair just three days to create Peeper Pam. "The most challenging bit was that we'd not used sockets – more efficient than the Pico constantly asking Raspberry Pi 'do you see anything?'," says Martin. "Raspberry Pi does all the heavy lifting, while Pico just listens for an 'I've seen something' signal."

While he notes that you could get Raspberry Pi 5 to serve both functions, the two-part setup means you can place the camera in a different position to monitor a spot you can't see. Also, by adapting the code from the project's GitHub repo (**magpi.cc/peeperpamgh**), there are lots of other uses if you get the AI to deter other objects. "Pigeons in the window box is one that we want to do," says Martin.

▼ The project comprises two parts: the AI KIT with Camera Module and server on Raspberry Pi 5, and the 'detector' powered by Pico W

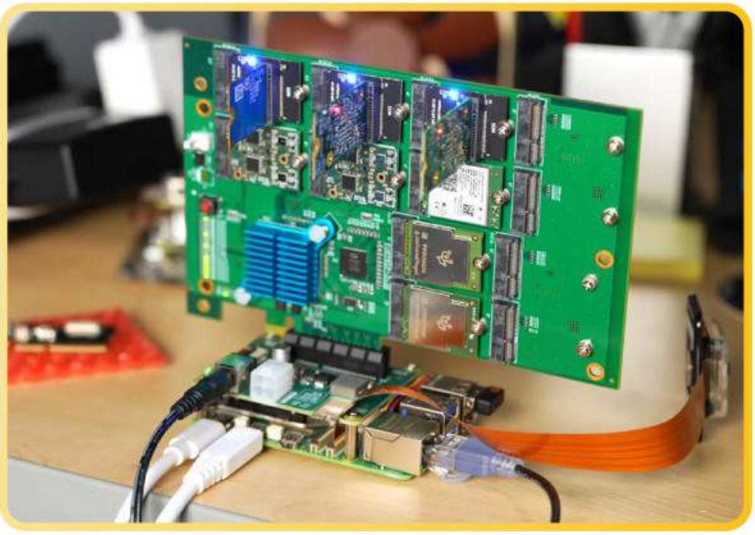




▲ When a person is detected in the camera video feed, the level of certainty is indicated by the detector's voltmeter

AI Kit projects continued

With eight neural processors offering a total of 55 TOPS, this truly is a monster of an AI PC



Monster AI Pi PC

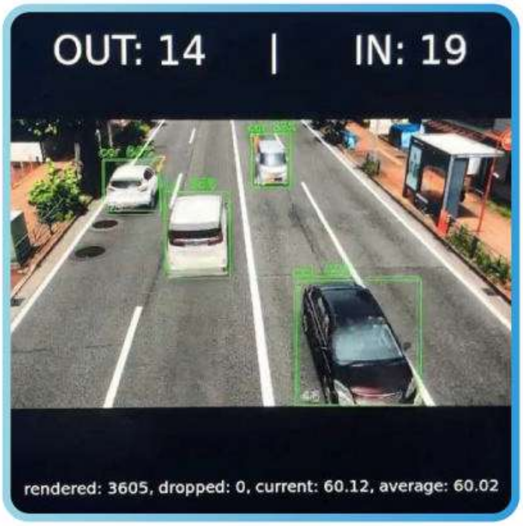
Jeff Geerling

magpi.cc/monsteraipi

Never one to do things by halves, Jeff Geerling went overboard with Raspberry Pi AI Kit and built a Monster AI Pi PC with a total of eight neural processors. In fact, with 55 TOPS (trillions of operations per second), it's faster than the latest AMD, Qualcomm and Apple Silicon processors!

The NPU chips – including the AI Kit's Hailo-8L – are connected to a large 12× PCIe slot card with a PEX 8619 switch capable of handling 16 PCI Express Gen 2 lanes. The card is then mounted on a Raspberry Pi 5 via a Pineboards uP City Lite HAT, which has an extra 12V PSU to supply the extra wattage needed for all those processors.

With a bit of jiggery-pokery with the firmware and drivers on Raspberry Pi, Jeff managed to get it working.



```
OUT: 14   |   IN: 19
```

```
rendered: 3605, dropped: 0, current: 60.12, average: 60.02
```

Computer vision is used to detect and count vehicles crossing a live camera view

Car Detection & Tracking System

Naveen

magpi.cc/cartrackingai

As a proof of concept, Japanese maker Naveen aimed to implement an automated system for identifying and monitoring cars at toll plazas to get an accurate tally of the vehicles entering and exiting.

With the extra processing power provided by a Raspberry AI Kit, the project uses Edge Impulse computer vision to detect and count cars in the view from a Camera Module Wide. "We opted for a wide lens because it can capture a larger area," he says, "allowing the camera to monitor multiple lanes simultaneously. He also needed to train and test a YOLOv5 machine learning model. All the details can be found on the project page via the link above – which could prove useful for learning how to train custom ML models for your own AI project.

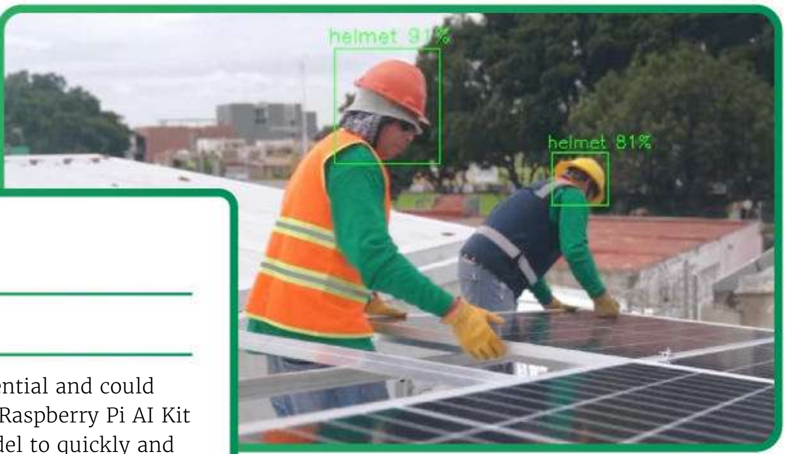
Safety Helmet Detection System

Shakhizat Nurgaliyev

magpi.cc/safetyhelmetai

Wearing a safety helmet on a building site is essential and could save your life. This computer vision project uses Raspberry Pi AI Kit with the advanced YOLOv8 machine learning model to quickly and accurately identify objects within the camera view, running at an impressive inference speed of 30fps.

The project page has a guide showing how to make use of Raspberry Pi AI Kit to achieve efficient AI inferencing for safety helmet detection. This includes details of the software installation and model training process, for which the maker has provided a link to a dataset of 5,000 images with bounding box annotations for three classes: helmet, person, and head.



▲ Based on the powerful YOLOv8 ML model, this project can detect whether workers are wearing helmets

Accelerating MediaPipe Models

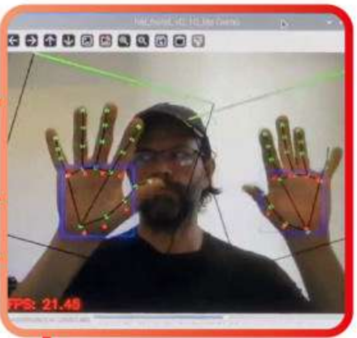
Mario Bergeron

magpi.cc/mediapipeaikit

Google's MediaPipe is an open-source framework developed for building machine learning pipelines, especially useful for working with videos and images.

Having used MediaPipe on other platforms, Mario Bergeron decided to experiment with it on a Raspberry Pi AI Kit. On the project page (linked above) he details the process, including using his Python demo application with options to detect hands/palms, faces, or poses.

Mario's test results show how much better the AI Kit's Hailo-8L AI accelerator module performs compared to running reference TensorFlow Lite models on Raspberry Pi 5 alone: up to 5.8 times faster. With three models running for hand and landmarks detection, the frame rate is 26–28fps with one hand detected; 22–25fps for two.



▲ MediaPipe models for hand, face, and pose detection run so much faster using the AI Kit

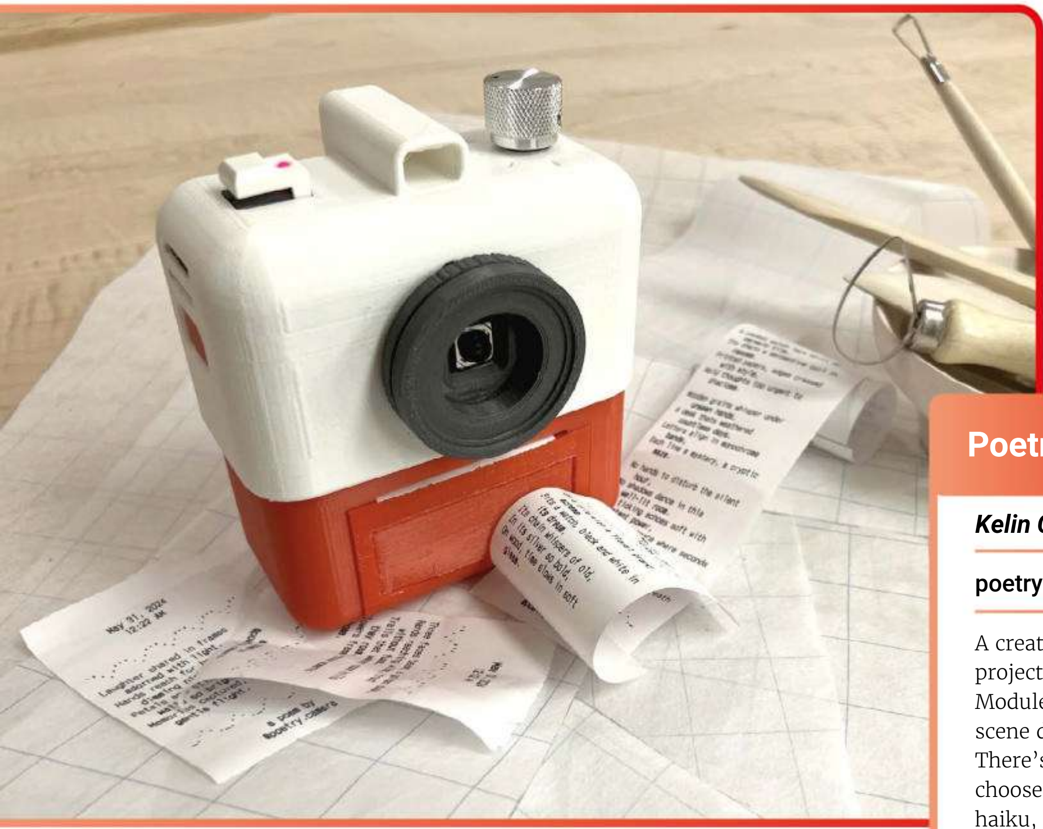
AI Hardware Options

While the AI Kit with its Hailo-8L module is a powerful add-on for enhancing your Raspberry Pi AI projects, there are a few other options. Google's Coral USB Accelerator is a handy plug-and-play device, while the Coral M.2 Accelerator with Dual Edge TPU supports parallel execution. An easy-to-use option for computer vision projects is the Vizy camera. AI is also built into some Raspberry Pi robots such as the XGO-mini2 and lite-2 dogs, and the ArmPi FPV AI Vision arm.

▲ The XGO-lite2 robotic dog has AI functions such as computer vision and speech recognition

More AI projects

Explore additional Raspberry Pi-powered AI creations



This very special camera produces printed poetic descriptions instead of pictures

> This ingenious art project turns the view from Camera Module 3 into a poetic description

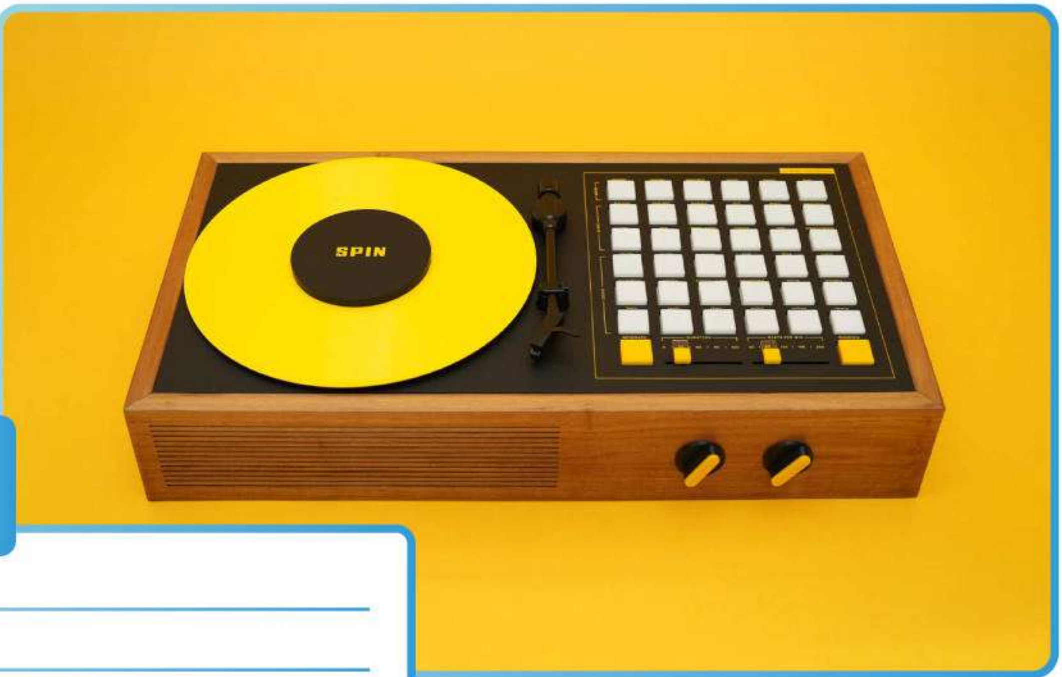
Poetry Camera

Kelin Carolyn Zhang and Ryan Mather

poetry.camera

A creative use of AI, this ingenious art project turns the view from Camera Module 3 into a poetic description of the scene delivered by a mini thermal printer. There's even a knob you can turn to choose different styles of poetry such as haiku, limerick, and sonnet.

Powered by a Raspberry Pi Zero 2 W in a 3D-printed case, the camera makes use of OpenAI's GPT-4 Vision API for its image-to-text capabilities. It does need a Wi-Fi connection to work, but you can always connect it to your mobile phone hotspot when out and about. If you want to make your own Poetry Camera, all the details and code are in the project's GitHub repo: **magpi.cc/poetrycamgh**.



This AI turntable generates intriguing musical combinations. How about some happy death-metal disco?

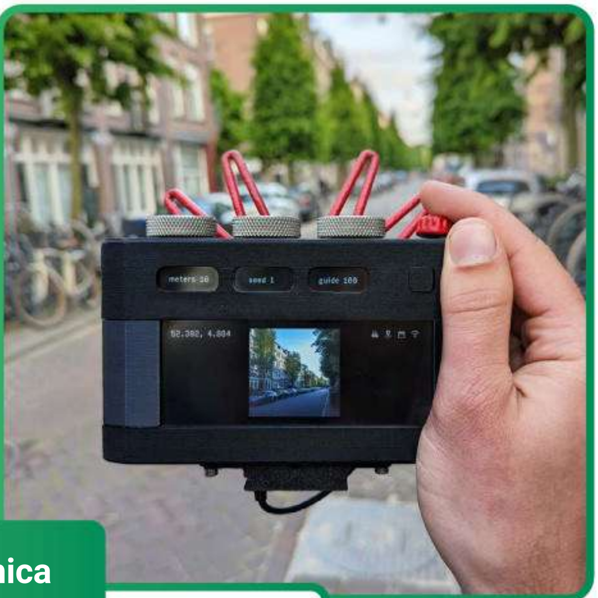
SPIN AI Music Synthesizer

Arvind Sanjeev

magpi.cc/spin

Seeing and hearing SPIN in action is astounding. Press the buttons to select any combination of options from three categories: mood, genre, and sound. Use a couple of sliders to set the duration and beats per minute, then generate the track using AI. To play it, you put the needle on the vinyl record – in true DJ style, you can even use your hand to slow it down, reverse it, and scratch.

Under the hood, SPIN's Raspberry Pi 4 prompts the MusicGen API to create an MP3 music file which is then loaded onto a Digital Vinyl System. A modified Numark PT-01 and time-coded control vinyl record serves as the turntable; using Xwax, Raspberry Pi reads the vinyl time-code through a Behringer audio driver, and the output is played via stereo speakers.



An AI image is produced based on the location and dial settings

Paragraphica

Bjørn Karmann

magpi.cc/paragraphica

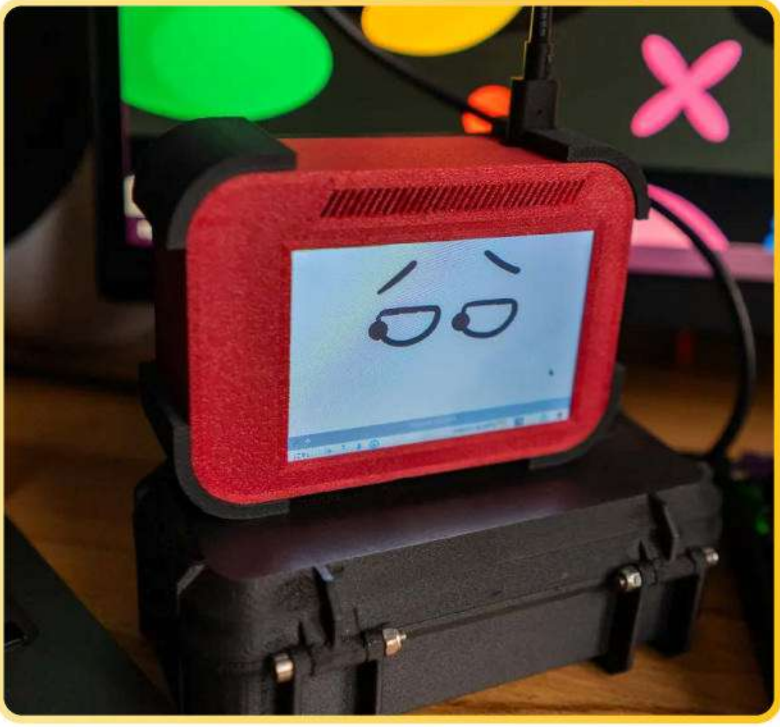
This strange-looking 'camera' doesn't take pictures. Instead, it generates images based on data for a location using open APIs, including OpenWeatherMap, and Mapbox. The data is used to compose a paragraph that details a representation of the current place and moment, and this description is then used as the AI image prompt.

Automatic Speech Recognition AI Assistant

Adam Frydrych

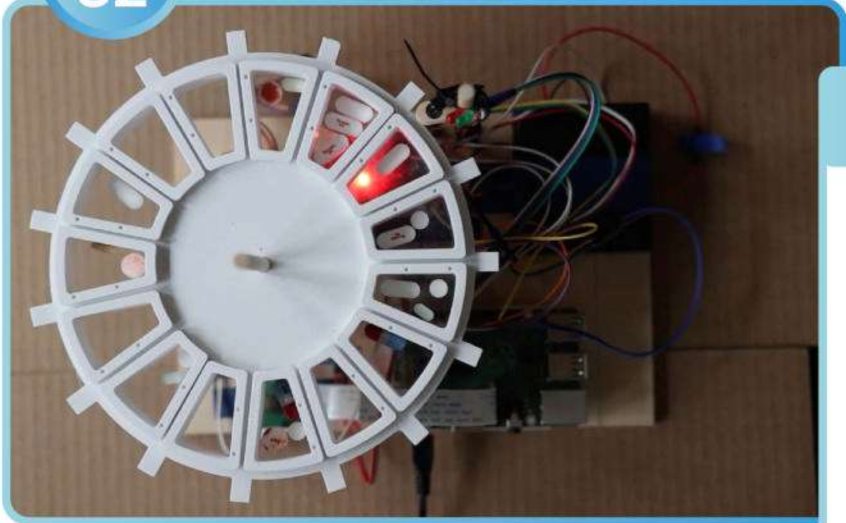
magpi.cc/autospeechrec

Housed in a neat 3D-printed case, this cute desktop assistant can understand spoken queries and display the answers on its LCD screen. A Raspberry Pi running VOSK voice recognition software communicates with an Ollama large language model hosted on a server PC. The use of NordVPN's Meshnet enables it to do that from any location.



Ask the assistant a question and it will show the answer on its LCD screen

Read more on page 32



This prototype pill dispenser uses AI to detect the status of containers in a spinning tray

Smart Pill Dispenser

Andrew Zhang

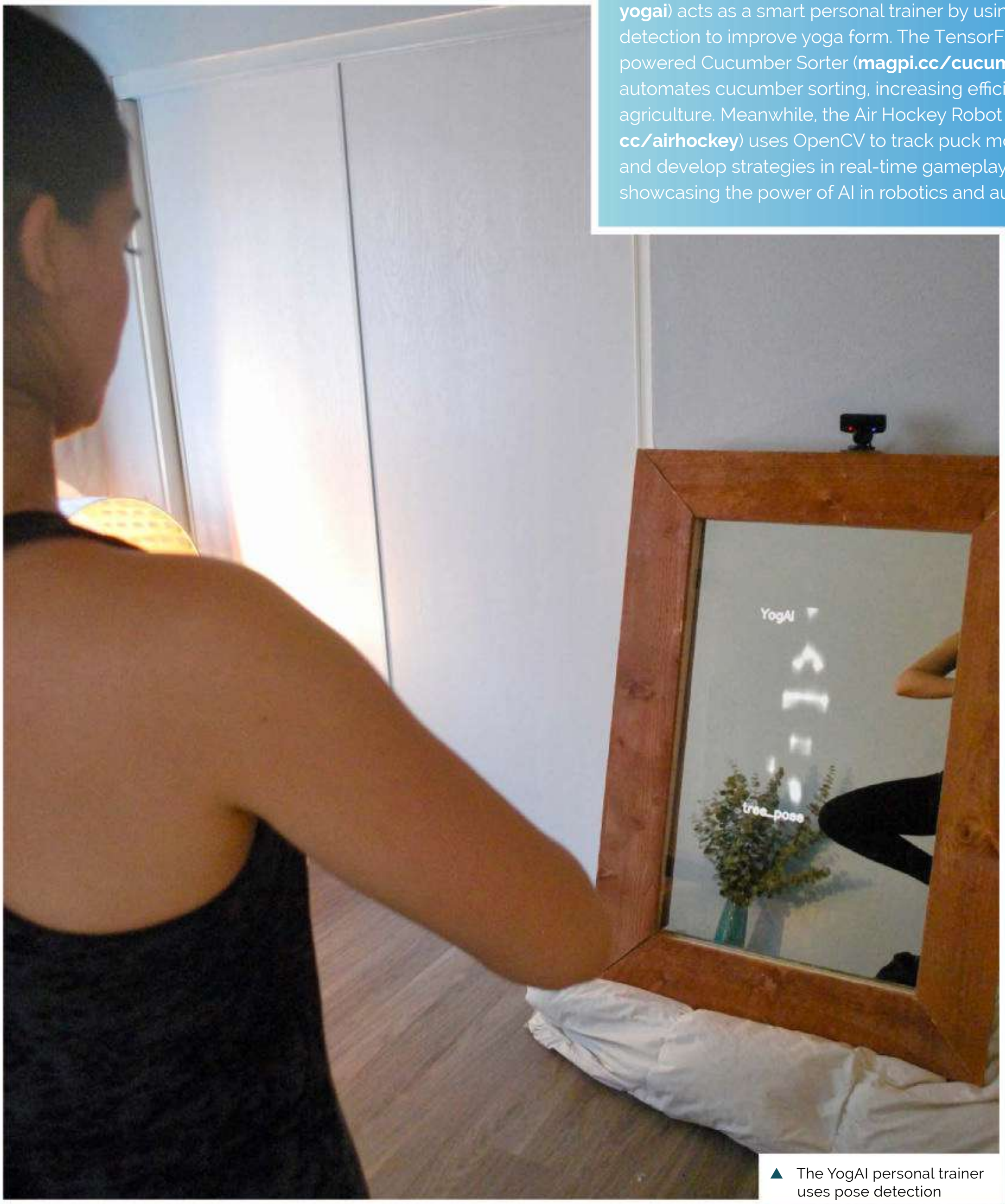
magpi.cc/smartpillai

A major issue in healthcare is patients forgetting to take their medication. This prototype smart pill dispenser offers a cost-effective solution. A Raspberry Pi-controlled stepper motor spins a circular tray with 15 containers for pills. Images from a Camera Module are analysed using a TensorFlow machine learning model to detect whether a container is full or empty. The patient and their doctor can log into a web server to check whether medication has been dispensed for each morning and afternoon.

Raspberry Pi has been used to power a wide range of computer vision projects

Computer vision

Raspberry Pi has been used to power a wide range of computer vision projects. For instance, VespAI (**magpi.cc/vespaigit**) helps to monitor invasive Asian hornets, supporting ecosystem protection. YogAI (**magpi.cc/yogai**) acts as a smart personal trainer by using pose detection to improve yoga form. The TensorFlow-powered Cucumber Sorter (**magpi.cc/cucumberai**) automates cucumber sorting, increasing efficiency in agriculture. Meanwhile, the Air Hockey Robot (**magpi.cc/airhockey**) uses OpenCV to track puck movement and develop strategies in real-time gameplay, showcasing the power of AI in robotics and automation.



▲ The YogAI personal trainer uses pose detection

Using the AI Camera

Get up and running with the new AI Camera. By **Nate Contino**

MAKER

Nate Contino

Nate is a retrofuturist and writes documentation for Raspberry Pi.

lambdalatitudinarians.org

You'll Need

- Raspberry Pi 4 Model B or Raspberry Pi 5
- Raspberry Pi AI camera **magpi.cc/aicamera**
- IMX500 Firmware

The Raspberry Pi AI Camera is an exciting new release. Here's how to run the pre-packaged MobileNet SSD and PoseNet neural network models on it.

Prerequisites

These instructions assume you are using the AI Camera attached to either a Raspberry Pi 4 Model B or Raspberry Pi 5 board. With minor changes, you can follow these instructions on other Raspberry Pi models with a camera connector, including the Raspberry Pi Zero 2 W and Raspberry Pi 3 Model B+.

First, ensure that your Raspberry Pi runs the latest software. Run the following command to update:

```
$ sudo apt update && sudo apt full-upgrade
```

Install the IMX500 firmware

The AI camera must download runtime firmware onto the IMX500 sensor during startup. To install these firmware files onto your Raspberry Pi, run the following command:

```
$ sudo apt install imx500-all
```

This command:

- Installs the **/lib/firmware/imx500_loader.fpk** and **/lib/firmware/imx500_firmware.fpk** firmware files required to operate the camera's IMX500 sensor
- Places a number of neural network model firmware files in **/usr/share/imx500-models/**
- Installs the IMX500 post-processing software stages in `rpicam-apps`
- And installs the Sony network model packaging tools

Take a break…

The IMX500 kernel device driver loads all the firmware files when the camera starts. This may take several minutes if the neural network model firmware has not been previously cached. The demos below display a progress bar on the console to indicate firmware loading progress.

Reboot

Now that you've installed the prerequisites, restart your Raspberry Pi:

```
$ sudo reboot
```

Run example applications

Once all the system packages are updated and firmware files installed, you can start running some example applications. As mentioned, the Raspberry Pi AI Camera integrates fully with libcamera, rpicam-apps, and Picamera2.

rpicam-apps

The rpicam-apps camera applications include IMX500 object detection and pose estimation stages that can be run in the post-processing pipeline. For more information about the post-processing pipeline, see the post-processing documentation (**magpi.cc/postprocess**).

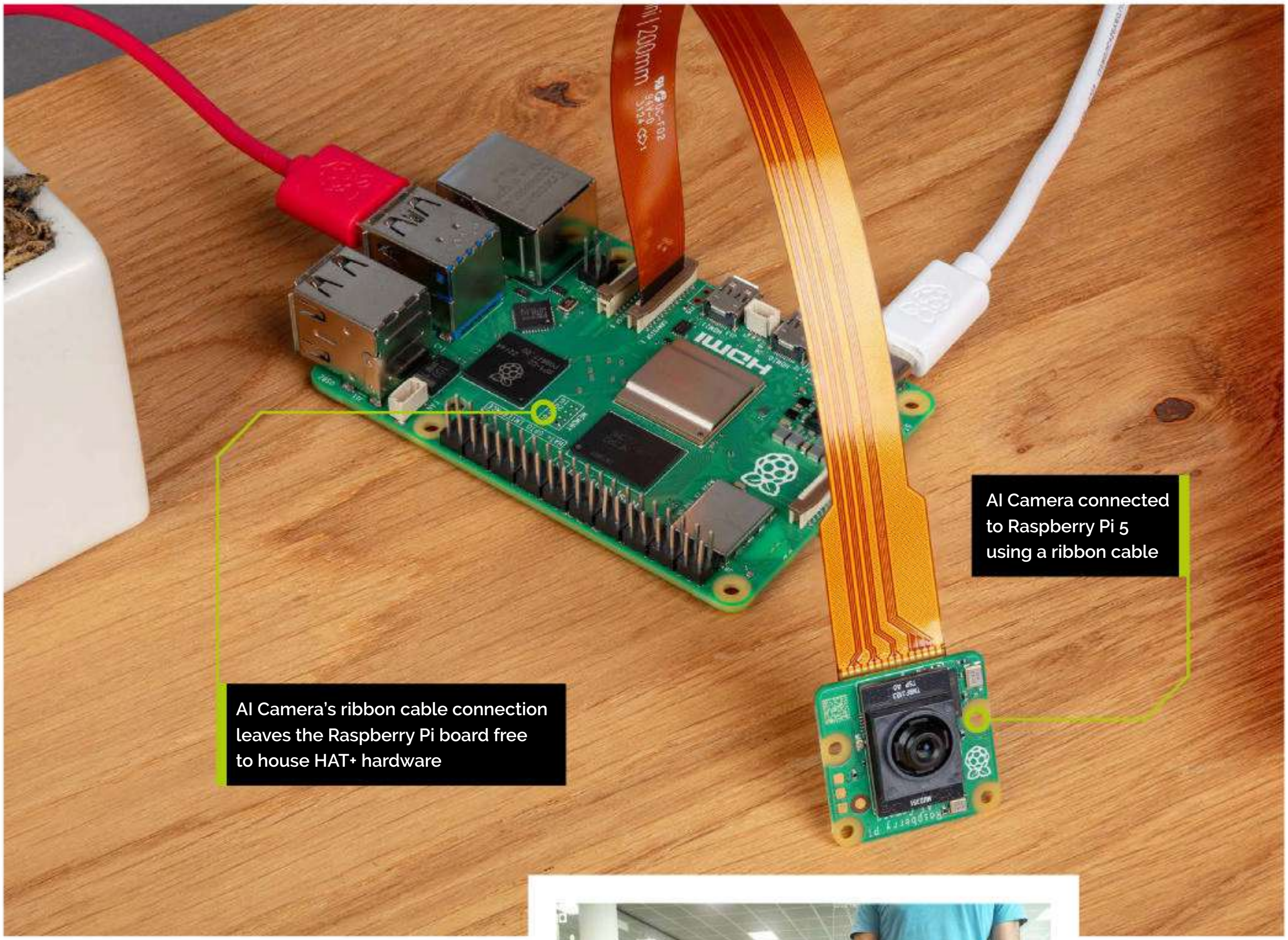
The examples on this page use post-processing JSON files located in **/usr/share/rpicam-assets/**.

Object detection

The MobileNet SSD neural network performs basic object detection, providing bounding boxes and confidence values for each object found. The file **imx500_mobilenet_ssd.json** contains the configuration parameters for the IMX500 object detection post-processing stage using the MobileNet SSD neural network.

imx500_mobilenet_ssd.json declares a post-processing pipeline that contains two stages:

1. `imx500_object_detection`, which picks out bounding boxes and confidence values generated by the neural network in the output tensor
2. `object_detect_draw_cv`, which draws bounding boxes and labels on the image



AI Camera connected to Raspberry Pi 5 using a ribbon cable

AI Camera's ribbon cable connection leaves the Raspberry Pi board free to house HAT+ hardware

The MobileNet SSD tensor requires no significant post-processing on your Raspberry Pi to generate the final output of bounding boxes. All object detection runs directly on the AI Camera.

The following command runs `rpicam-hello` with object detection post-processing:

```
$ rpicam-hello -t 0s --post-process-file /usr/
share/rpi-camera-assets/imx500_mobilenet_ssd.
json --viewfinder-width 1920 --viewfinder-
height 1080 --framerate 30
```

After running the command, you should see a viewfinder that overlays bounding boxes on objects recognised by the neural network (see **Figure 1**).

To record video with object detection overlays, use `rpicam-vid` instead. The following command runs `rpicam-hello` with object detection post-processing:

```
$ rpicam-vid -t 10s -o output.264 --post-
process-file /usr/share/rpi-camera-assets/
imx500_mobilenet_ssd.json --width 1920
--height 1080 --framerate 30
```

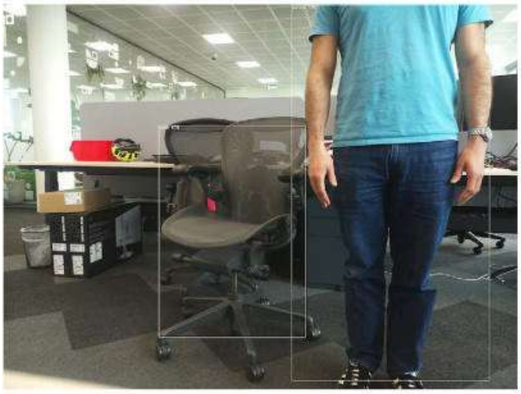


Figure 1: Object detection

You can configure the `imx500_object_detection` stage in many ways.

For example, `max_detections` defines the maximum number of objects that the pipeline will detect at any given time. `threshold` defines the minimum confidence value required for the pipeline to consider any input as an object.

The raw inference output data of this network can be quite noisy, so this stage also preforms some temporal filtering and applies hysteresis. To disable this filtering, remove the `temporal_filter` config block.

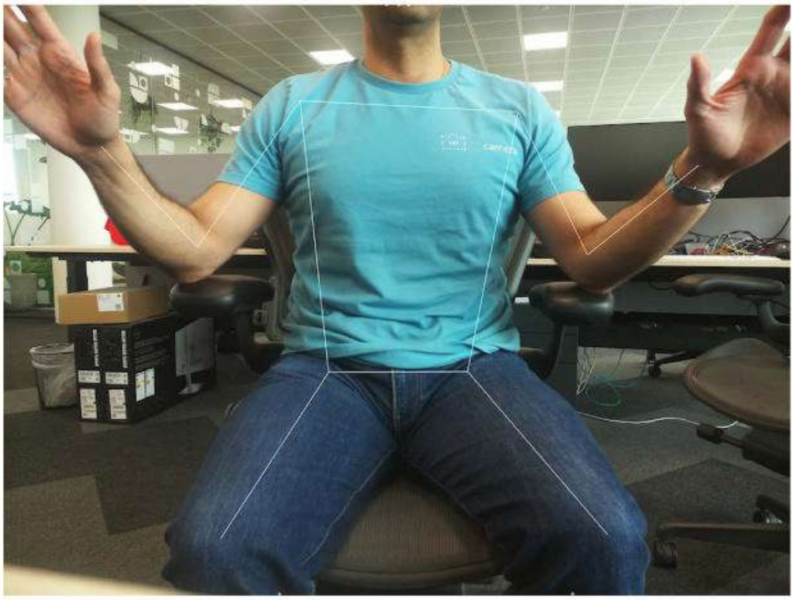


Figure 2: Pose detection

Pose estimation

The PoseNet neural network performs pose estimation, labelling key points on the body associated with joints and limbs. **imx500_posenet.json** contains the configuration parameters for the IMX500 pose estimation stage using the PoseNet neural network. **imx500_posenet.json** declares a post-processing pipeline that contains two stages:

1. `imx500_posenet`, which fetches the raw output tensor from the PoseNet neural network
2. `plot_pose_cv`, which draws line overlays on the image

The AI Camera performs basic detection, but the output tensor requires additional post-processing on your host Raspberry Pi to produce final output.

The following command runs `rpicam-hello` with pose estimation post-processing (**Figure 2**):

```
$ rpicam-hello -t 0s --post-process-file /usr/
share/rpi-camera-assets/imx500_posenet.json
--viewfinder-width 1920 --viewfinder-height
1080
```

You can configure the `imx500_posenet` stage in many ways. For example, `max_detections` defines the maximum number of bodies that the pipeline will detect at any given time. `threshold` defines the minimum confidence value required for the pipeline to consider input as a body.

Top Tip

Device drivers

The Raspberry Pi AI Camera Module documentation has more information on the device drivers used by AI Camera. **magpi.cc/aicamdrivers**

> The Raspberry Pi processes the images and then performs AI inference

Picamera2

For examples of image classification, object detection, object segmentation, and pose estimation using Picamera2, see the picamera2 GitHub repository (**magpi.cc/picamera2**).

Most of the examples use OpenCV for some additional processing. To install the dependencies required to run OpenCV, run the following command:

```
$ sudo apt install python3-opencv python3-
munkres
```

Now download the the picamera2 repository to your Raspberry Pi to run the examples. You'll find example files in the root directory, with additional information in the **README.md** file.

Run the following script from the repository to run YOLOv8 object detection:

```
$ python imx500_object_detection_demo.py
--model /usr/share/imx500-models/imx500_
network_yolov8n_pp.rpk --ignore-dash-labels
-r
```

To try pose estimation in Picamera2, run the following script from the repository:

```
$ python imx500_pose_estimation_higherhrnet_
demo.py
```

Under the Hood

The Raspberry Pi AI Camera works differently from traditional AI-based camera image processing systems, as shown in **Figure 3.**

The left side demonstrates the architecture of a traditional AI camera system. In such a system, the camera delivers images to the Raspberry Pi. The Raspberry Pi processes the images and then performs AI inference. Traditional systems may use external AI accelerators (as shown) or rely exclusively on the CPU.

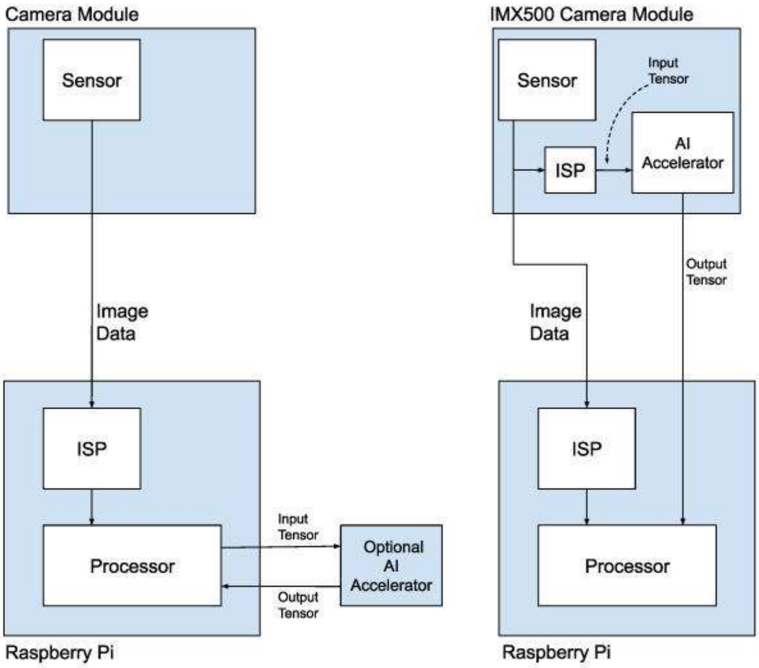


Figure 3: Camera Module vs AI Camera Module

Figure 4: The various software components (green) and Raspberry Pi hardware components (red)

The right side demonstrates the architecture of a system that uses IMX500. The camera module contains a small Image Signal Processor (ISP) which turns the raw camera image data into an input tensor. The camera module sends this tensor directly into the AI accelerator within the camera, which produces an output tensor that contains the inferencing results. The AI accelerator sends this tensor to the Raspberry Pi. There is no need for an external accelerator, nor for the Raspberry Pi to run neural network software on the CPU.

To fully understand this system, familiarise yourself with the following concepts:

Input Tensor

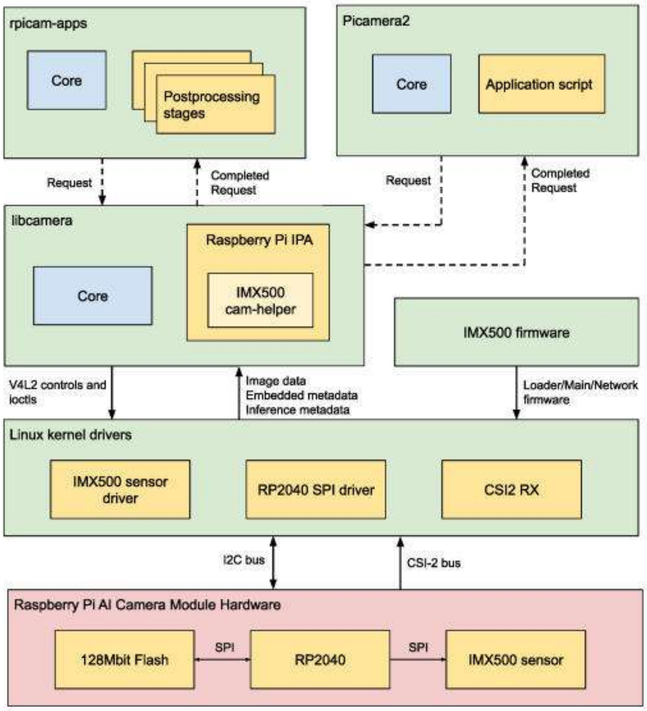
The part of the sensor image passed to the AI engine for inferencing. Produced by a small on-board ISP which also crops and scales the camera image to the dimensions expected by the neural network that has been loaded. The input tensor is not normally made available to applications, though it is possible to access it for debugging purposes.

Region of Interest (ROI)

Specifies exactly which part of the sensor image is cropped out before being rescaled to the size demanded by the neural network. Can be queried and set by an application. The units used are always pixels in the full resolution sensor output. The default ROI setting uses the full image received from the sensor, cropping no data.

Output Tensor

The results of inferencing performed by the neural network. The precise number and shape of the outputs depend on the neural network. Application code must understand how to handle the tensor.



System Architecture

The diagram in **Figure 4** shows the various camera software components (in green) used during our imaging/inference use case with the Raspberry Pi AI Camera module hardware (in red):

At startup, the IMX500 sensor module loads firmware to run a particular neural network model. During streaming, the IMX500 generates both an image stream and an inference stream. This inference stream holds the inputs and outputs of the neural network model, also known as input/output tensors.

Deep dive into Raspberry Pi config.txt

Configure your Raspberry Pi like a professional.
By **Nate Contino**

MAKER

Nate Contino

Nate is a retrofuturist and writes documentation for Raspberry Pi.

lambdalatitudinarians.org

You'll Need

- Raspberry Pi
- Raspberry Pi OS

Instead of the BIOS found on a conventional PC, Raspberry Pi devices use a configuration file called config.txt. The GPU reads this file before the Arm CPU and Linux initialise. Raspberry Pi OS looks for this file in the boot partition, located at **/boot/firmware/**.

You can edit this file using Nano (**magpi.cc/nano**). Open a Terminal window and enter:

```
sudo nano /boot/firmware/config.txt
```

… or to use Vi (**magpi.cc/vi**):

```
sudo vi /boot/firmware/config.txt
```

In this tutorial we'll take a look at some of the configuration settings you can adjust in the default **config.txt** file. There are many further configuration settings available. For more detailed information on settings see our Raspberry Pi documentation page (**magpi.cc/config.txt**)

You can edit **config.txt** directly from your Raspberry Pi OS installation using Nano, Vi or another text editor. You can also remove the storage device and edit files in the boot partition, including **config.txt**, from a separate computer.

Changes to **config.txt** only take effect after a reboot:

```
sudo reboot
```

You can view the current active settings using the following commands:

```
vcgencmd get_config <config>
```

…displays a specific config value, e.g. `vcgencmd get_config arm_freq`

```
vcgencmd get_config int
```

…lists all non-zero integer config options (non-zero), and

```
vcgencmd get_config str
```

…lists all non-null string config options.

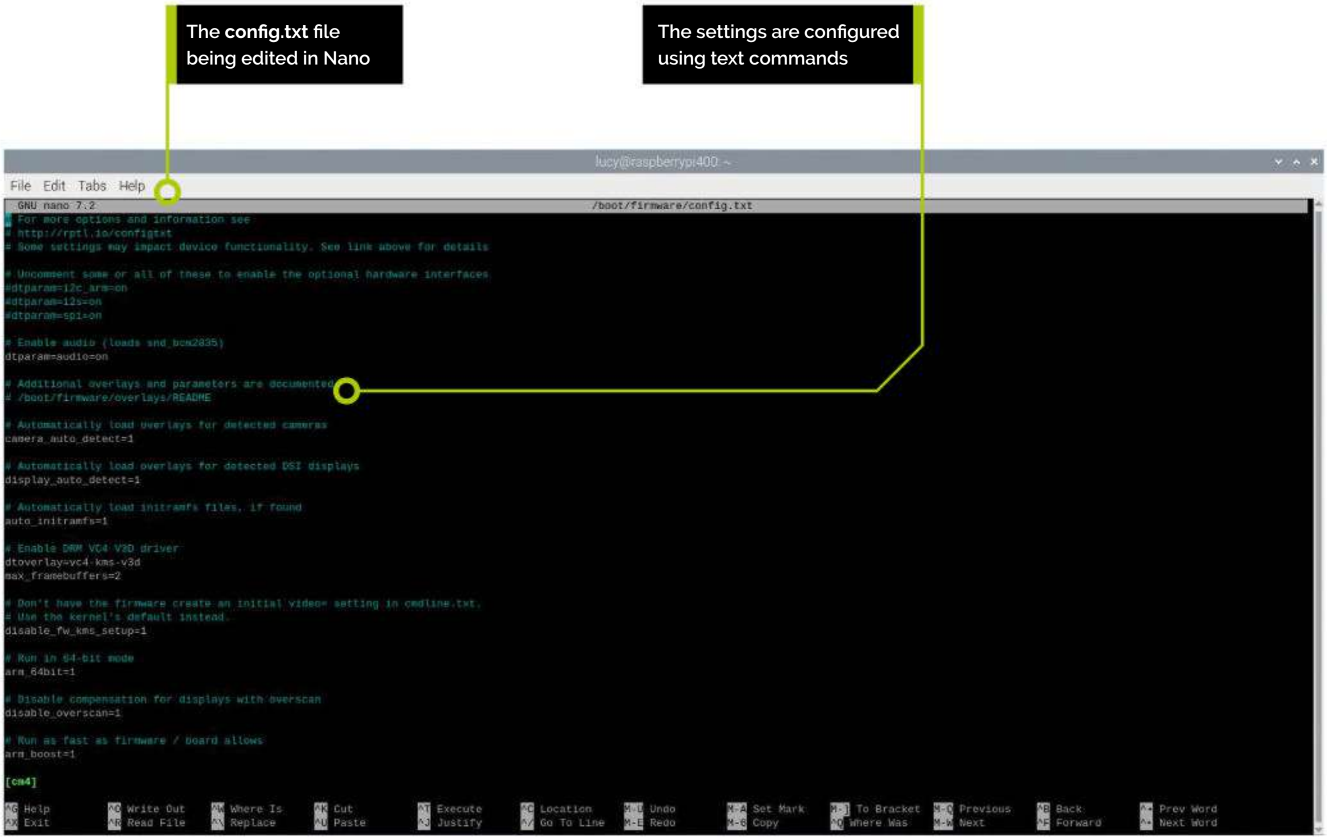
File format

The **config.txt** file is read by the early-stage boot firmware, so it uses a very simple file format: a single `property=value` statement on each line, where value is either an integer or a string. Comments may be added, or existing config values may be commented out and disabled, by starting a line with the `#` character.

There is a 98-character line length limit for entries. Raspberry Pi OS ignores any characters past this limit. See the **config.txt** code on p56 to view the default file.

Include an extra file

The `include` command causes the content of the specified file to be inserted into the current file For example, adding `include extraconfig.txt` to **config.txt** will include the content of the **extraconfig.txt** file in the **config.txt** file.



The config.txt file being edited in Nano

The settings are configured using text commands

```
  GNU nano 7.2                                        /boot/firmware/config.txt
# For more options and information see
# http://rptl.io/configtxt
# Some settings may impact device functionality. See link above for details

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# Additional overlays and parameters are documented
# /boot/firmware/overlays/README

# Automatically load overlays for detected cameras
camera_auto_detect=1

# Automatically load overlays for detected DSI displays
display_auto_detect=1

# Automatically load initramfs files, if found
auto_initramfs=1

# Enable DRM VC4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2

# Don't have the firmware create an initial video= setting in cmdline.txt.
# Use the kernel's default instead.
disable_fw_kms_setup=1

# Run in 64-bit mode
arm_64bit=1

# Disable compensation for displays with overscan
disable_overscan=1

# Run as fast as firmware / board allows
arm_boost=1

[cm4]

^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket  M-Q Previous  ^B Back     ^◂ Prev Word
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line  M-E Redo  M-6 Copy      ^Q Where Was    M-W Next      ^F Forward  ^▸ Next Word
```

Parameters

Device Tree configuration files for Raspberry Pi support a number of parameters for such things as enabling I2C and SPI interfaces. Many DT overlays are configurable via the use of parameters. Both types of parameters can be supplied using the `dtparam` setting. In addition, overlay parameters can be appended to the `dtoverlay` option, separated by commas, but keep in mind the line length limit of 98 characters.

camera_auto_detect

With this setting enabled (in Raspberry Pi OS it is enabled by default), the firmware will automatically load overlays for CSI cameras that it recognises. Set `camera_auto_detect=0` to disable the setting.

display_auto_detect

With this setting enabled (it is enabled by default in Raspberry Pi OS), the firmware will automatically load overlays for DSI displays that it recognises. Set `display_auto_detect=0 to` disable.

auto_initramfs

The default `auto_initramfs=1` instructs Raspberry Pi OS to look for an **initramfs** file using the same rules as the kernel selection.

initramfs

The `initramfs` command specifies both the ramfs filename and the memory address to which to load it. It performs the actions of both `ramfsfile` and `ramfsaddr` in one parameter. The address can also be `followkernel` (or `0`) to place it in memory after the kernel image. Example values are: `initramfs initramf.gz 0x00800000` or `initramfs init.gz followkernel`. As with `ramfsfile`, newer firmwares allow the loading of multiple files by comma-separating their names.

This option uses different syntax from all the other options, and you should not use the `=` character here.

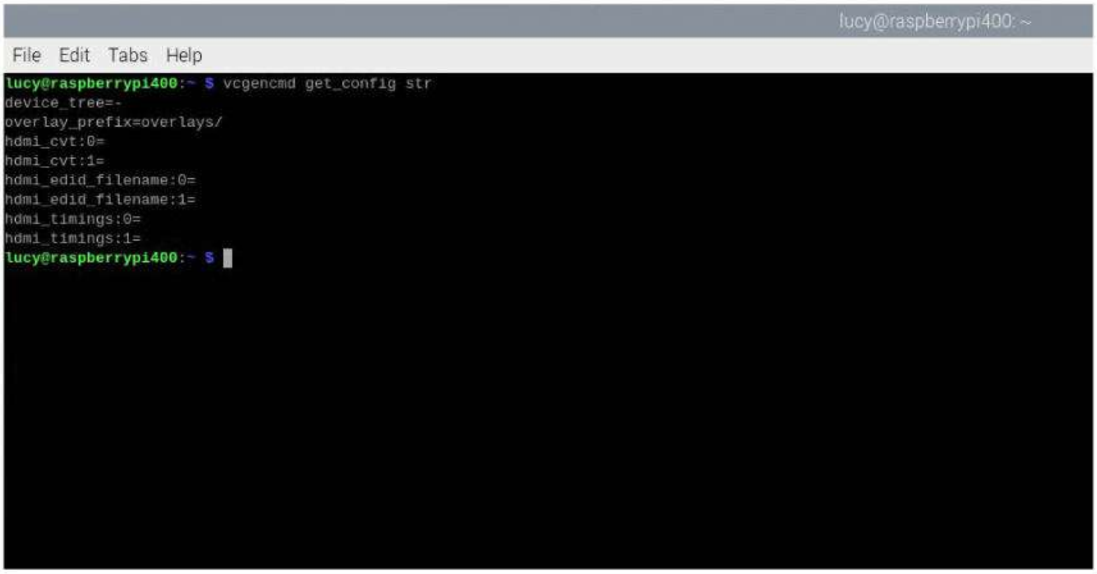
boot_partition

Specifies the partition number for booting unless the partition number was already specified

Top Tip

Note

Prior to Raspberry Pi OS Bookworm, Raspberry Pi OS stored the boot partition at **/boot/**. Here you will still find a **config.txt** file but the text redirects you to the **/boot/firmware/** file directory.



```
lucy@raspberrypi400:~ $ vcgencmd get_config str
device_tree=-
overlay_prefix=overlays/
hdmi_cvt:0=
hdmi_cvt:1=
hdmi_edid_filename:0=
hdmi_edid_filename:1=
hdmi_timings:0=
hdmi_timings:1=
lucy@raspberrypi400:~ $ 
```

Using vcgencmd to view the string-based configuration commands available

as a parameter to the reboot command (e.g. `sudo reboot 2`).

Partition numbers start at 1, and the MBR partitions are 1 to 4. Specifying partition 0 means boot from the default partition which is the first bootable FAT partition.

Bootable partitions must be formatted as FAT12, FAT16 or FAT32 and contain a **start.elf** file (or **config.txt** file on Raspberry Pi 5) in order to be classed as be bootable by the bootloader.

max_framebuffers

This configuration entry sets the maximum number of firmware framebuffers that can be created. Valid options are 0, 1, and 2. By default on devices before the Raspberry Pi 4 this is set to 1, so will need to be increased to 2 when using more than one display, for example HDMI and a DSI or DPI display. The Raspberry Pi 4 configuration sets this to 2 by default as it has two HDMI ports.

It is safe to set this to 2 in most instances, as framebuffers will only be created when an attached device is actually detected.

Setting this value to 0 can be used to reduce memory requirements when used in headless mode, as it will prevent any framebuffers from being allocated.

disable_fw_kms_setup

By default, the firmware parses the EDID of any HDMI attached display, picks an appropriate video mode, then passes the resolution and frame rate of the mode (and overscan parameters) to the Linux kernel via settings on the kernel command line. In rare circumstances, the firmware can choose a mode not in the EDID that may be incompatible with the device. Use `disable_fw_kms_setup=1` to disable passing video mode parameters, which can avoid this problem. The Linux video mode system (KMS) instead parses the EDID itself and picks an appropriate mode.

On Raspberry Pi 5, this parameter defaults to 1.

> “If arm_64bit=1, the kernel will be started in 64-bit mode”

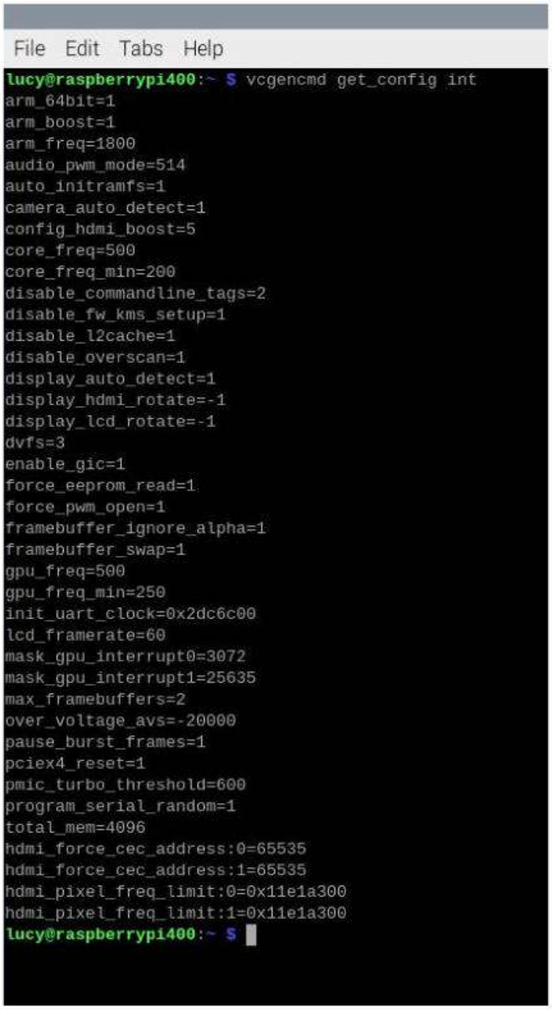
Arm 64bit

If `arm_64bit=1`, the kernel will be started in 64-bit mode. Setting to `0` selects 32-bit mode.

In 64-bit mode, the firmware will choose an appropriate kernel (e.g. **kernel8.img**), unless there is an explicit kernel option defined, in which case that is used instead.

Defaults to 1 on Pi 4s (Pi 4B, Pi 400, CM4 and CM4S), and 0 on all other platforms. However, if the name given in an explicit kernel option matches one of the known kernels, then `arm_64bit` will be set accordingly.

64-bit kernels may be uncompressed image files or a gzip archive of an image (which can still be



```
lucy@raspberrypi400:~ $ vcgencmd get_config int
arm_64bit=1
arm_boost=1
arm_freq=1800
audio_pwm_mode=514
auto_initramfs=1
camera_auto_detect=1
config_hdmi_boost=5
core_freq=500
core_freq_min=200
disable_commandline_tags=2
disable_fw_kms_setup=1
disable_l2cache=1
disable_overscan=1
display_auto_detect=1
display_hdmi_rotate=-1
display_lcd_rotate=-1
dvfs=3
enable_gic=1
force_eeprom_read=1
force_pwm_open=1
framebuffer_ignore_alpha=1
framebuffer_swap=1
gpu_freq=500
gpu_freq_min=250
init_uart_clock=0x2dc6c00
lcd_framerate=60
mask_gpu_interrupt0=3072
mask_gpu_interrupt1=25635
max_framebuffers=2
over_voltage_avs=-20000
pause_burst_frames=1
pciex4_reset=1
pmic_turbo_threshold=600
program_serial_random=1
total_mem=4096
hdmi_force_cec_address:0=65535
hdmi_force_cec_address:1=65535
hdmi_pixel_freq_limit:0=0x11e1a300
hdmi_pixel_freq_limit:1=0x11e1a300
lucy@raspberrypi400:~ $ 
```

◀ Using vcgencmd to view the integer-based configuration command list

called **kernel8.img**; the bootloader will recognise the archive from the signature bytes at the beginning). The 64-bit kernel will only work on the Raspberry Pi 3, 3+, 4, 400, Zero 2 W and 2B rev 1.2, and Raspberry Pi Compute Modules 3, 3+ and 4. Raspberry Pi 5 only supports a 64-bit kernel, so this parameter has been removed for that device.

arm_boost (Raspberry Pi 4 Only)

All Raspberry Pi 400s and newer revisions of the Raspberry Pi 4B are equipped with a second switch-mode power supply for the SoC voltage rail, and this allows the default turbo-mode clock to be increased from 1.5GHz to 1.8GHz. This change is enabled by default in Raspberry Pi OS. Set `arm_boost=0` to disable.

otg_mode (Raspberry Pi 4 only)

USB On-The-Go (often abbreviated to OTG) is a feature that allows supporting USB devices with an appropriate OTG cable to configure themselves as USB hosts. On older Raspberry Pi computers, a single USB 2 controller was used in both USB host and device mode.

Raspberry Pi 4B and Raspberry Pi 400 (not CM4 or CM4IO) add a high-performance USB 3 controller, attached via PCIe, to drive the main USB ports. The legacy USB 2 controller is still available on the USB-C power connector for use as a device (`otg_mode=0`, the default).

`otg_mode=1` requests that a more capable XHCI USB 2 controller is used as another host controller on that USB-C connector.

Because CM4 and CM4IO don't include the external USB 3 controller, Raspberry Pi OS images set `otg_mode=1` on CM4 for better performance.

power_force_3v3_pwm (Raspberry Pi 5 Only)

Forces PWM when using a 3V3 power supply. Set `power_force_3v3_pwm=0` to disable.

dtoverlay

The `dtoverlay` option requests the firmware to load a named Device Tree overlay – a configuration file that can enable kernel support for built-in and external hardware. For example, `dtoverlay=vc4-kms-v3d` loads an overlay that enables the kernel graphics driver.

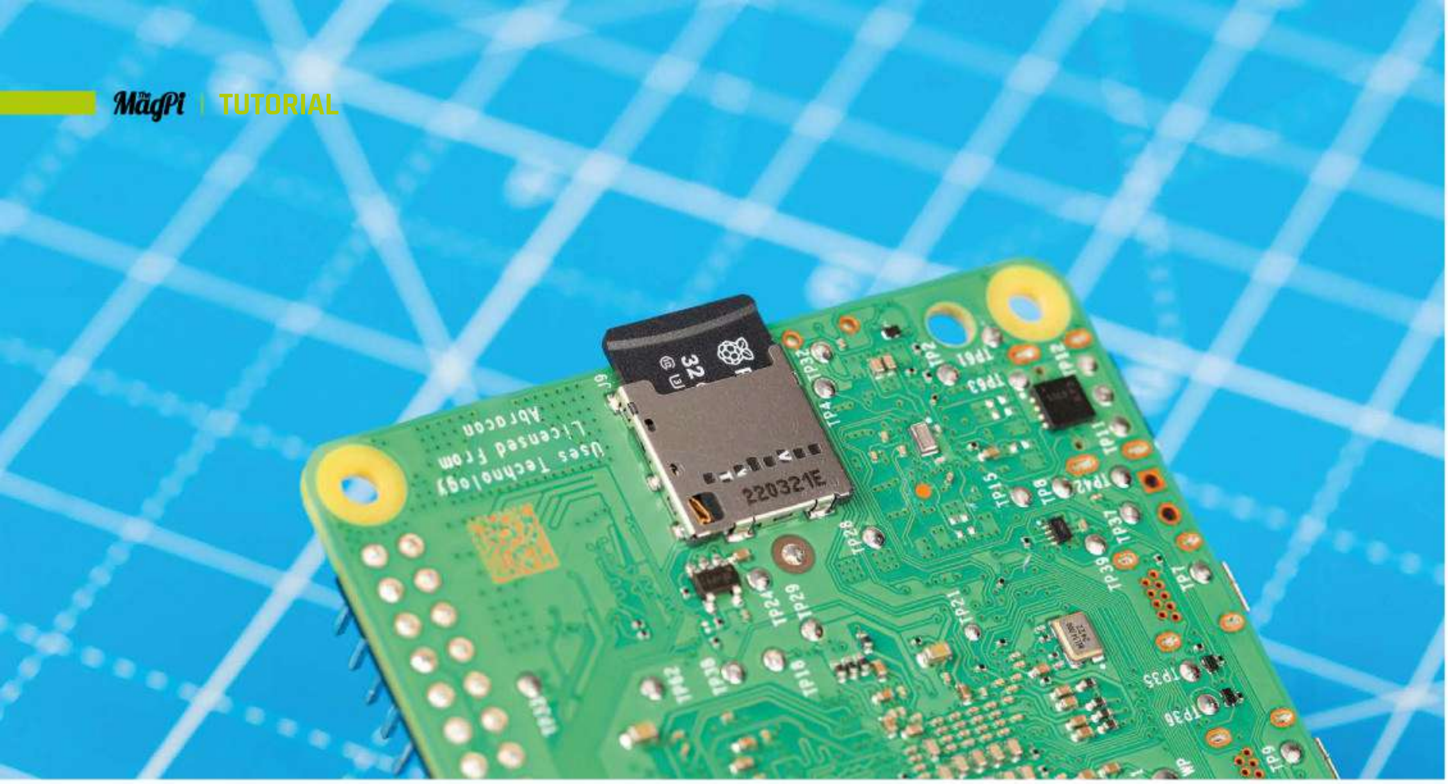
As a special case, if called with no value – `dtoverlay=` – the option marks the end of a list of overlay parameters. If used before any other dtoverlay or dtparam setting, it prevents the loading of any HAT overlay.

For more details, see the appropriate section of the online documentation at **magpi.cc/dtbs**.

Top Tip

Retrieval

Not all config settings can be retrieved using vcgencmd.



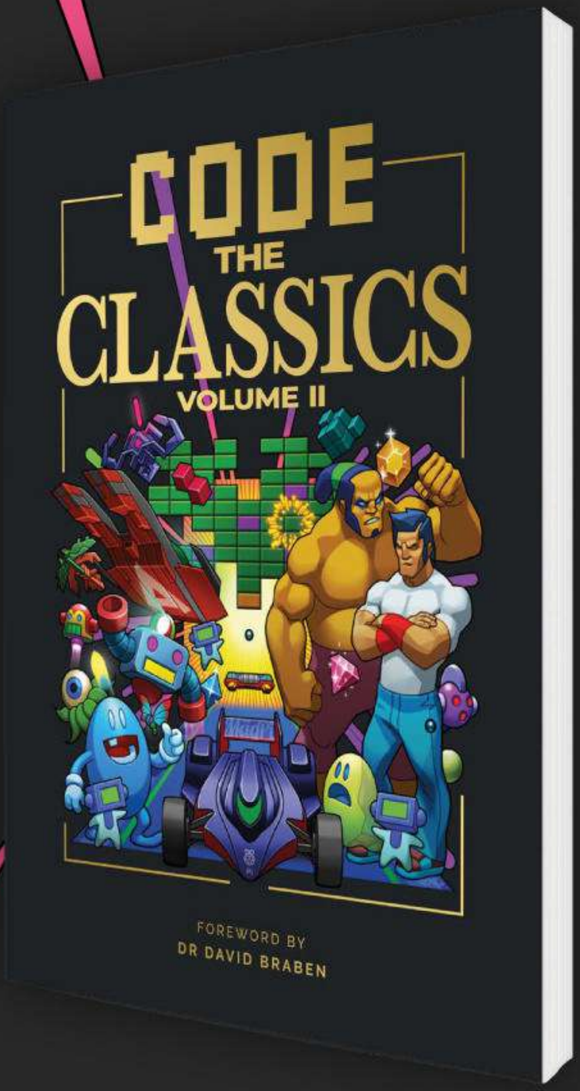
config.txt

Language: **Text**

DOWNLOAD THE FULL CODE:

magpi.cc/configtxt

```
# For more options and information see
# http://rptl.io/configtxt
# Some settings may impact device functionality. See link above for details

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# Additional overlays and parameters are documented
# /boot/firmware/overlays/README

# Automatically load overlays for detected cameras
camera_auto_detect=1

# Automatically load overlays for detected DSI displays
display_auto_detect=1

# Automatically load initramfs files, if found
auto_initramfs=1

# Enable DRM VC4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2

# Don't have the firmware create an initial video= setting in cmdline.txt.
# Use the kernel's default instead.
disable_fw_kms_setup=1

# Run in 64-bit mode
arm_64bit=1

# Disable compensation for displays with overscan
disable_overscan=1

# Run as fast as firmware / board allows
arm_boost=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[cm5]
dtoverlay=dwc2,dr_mode=host

[all]
```



CODE
THE
CLASSICS
VOLUME II
FOREWORD BY
DR DAVID BRABEN

HackSpace

TECHNOLOGY IN YOUR HANDS

HACK | MAKE | BUILD | CREATE

TOP PROJECTS

OBJET 3D'ART

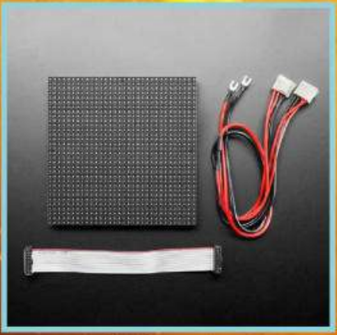
FORGE

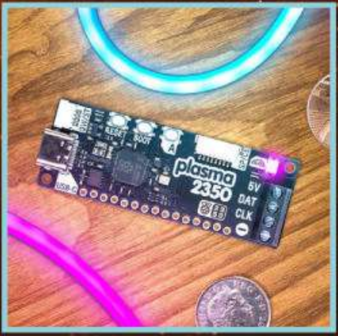
PG 92

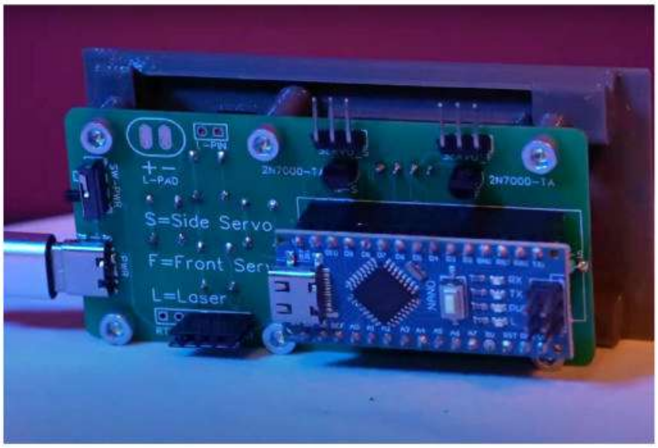
ALL THINGS LED

Everything you need to make your projects shiny and bright

Best of Breed







Gatoino

By **Small Batch Factory**

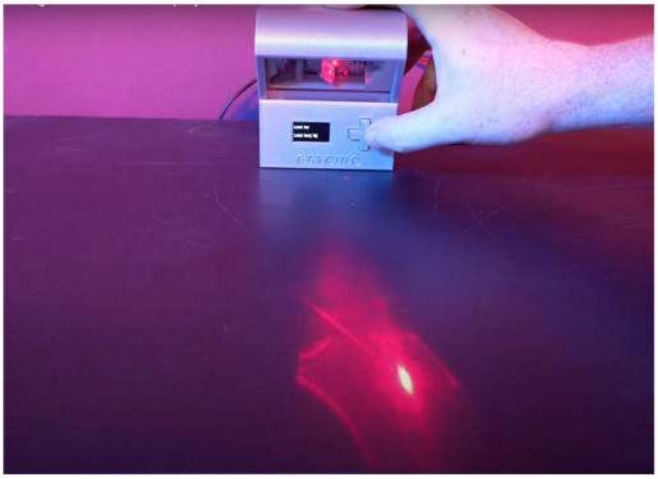
hsmag.cc/Gatoino

Cats are big business. They're insanely popular, with 20% of households in most western countries owning at least one form of cat. Their owners love them too, and will spend money on devices to make them happy. Devices such as this Gatoino, by Small Batch Factory.

Cats love laser pointers, and humans love automating things, so the Gatoino was born: a truly random, automatic laser pointer for cats. The development process for this device stretches over years – at first it was a device made from an off-the-shelf mechanism and a couple of servos, but over years of development overseen by a small team of feline experts, the Gatoino is now a fully custom piece of kit.

You can buy automatic laser pointers in the shops, but they usually follow the same simple pattern over and over again. Worse, some of them have to be put on the ground to work, making them an obstacle for humans, or easily knock-overable by the cats who are playing with them. The Gatoino incorporates a timer, to set the duration and interval between each play session, and you can even control the brightness of the laser and the limits of the laser travel, so you can restrict your cat's play area to keep them away from your collection of priceless antiques.

According to the maker, all cats deserve to have fun, so he's offering the build plans, software, 3D printing model and PCB files for sale. The BOM comes to about $35 (£27); that includes common parts like resistors and headers that you probably have already, so the true cost will be a bit below that. □



Right As the name suggests, the Gatoino uses an Arduino Nano to control a laser pointer to torment/delight your feline friends

Warning!
Laser pointer

Be careful when using laser pointers in your projects and don't aim a laser pointer at a person's head.

magpi.cc/lasersafety

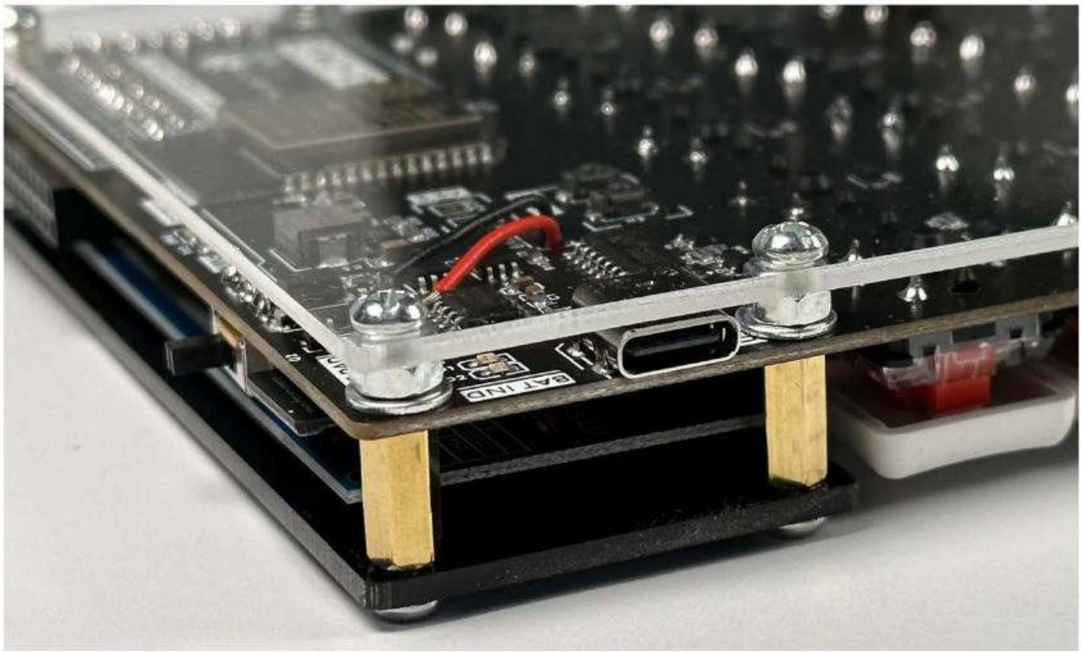
Sci-calc

By **Shao Duan** hsmag.cc/MechanicalSwitchCalculator

Mechanical keyboards are one of those things that enthusiasts spend a lot of time and effort thinking about. As with anything that offers a high degree of customisation, they're replaced, upgraded and argued over more than is healthy, and there's a brilliant, unbelievably creative scene of DIY makers creating their own keyboard designs. So it's a mystery really why we haven't seen a scientific calculator with mechanical keys until now.

This example, by Shao Duan, is a four-in-one machine, comprising a scientific calculator, a Bluetooth macropad, a handheld game console and a development board. It's powered by an ESP32, uses a 256x64 OLED screen, and uses clicky Kailh choc switches with custom 3D-printed keycaps. □

Right Many teachers have promised it, but looks like this device might actually Make Maths Fun



esc
÷
(
×
)
-
7
Home
8
↑
9
PgUp
+
sin
asin
sec
4
←
5
6
→
cos
acos
csc
2
↓
xʸ
√x
sel
tan
atan
3
PgDn
back
cot
0
.
=
eˣ
ln
RPN
Norm
del
ang
tab

Raspberry Pi Zero laptop

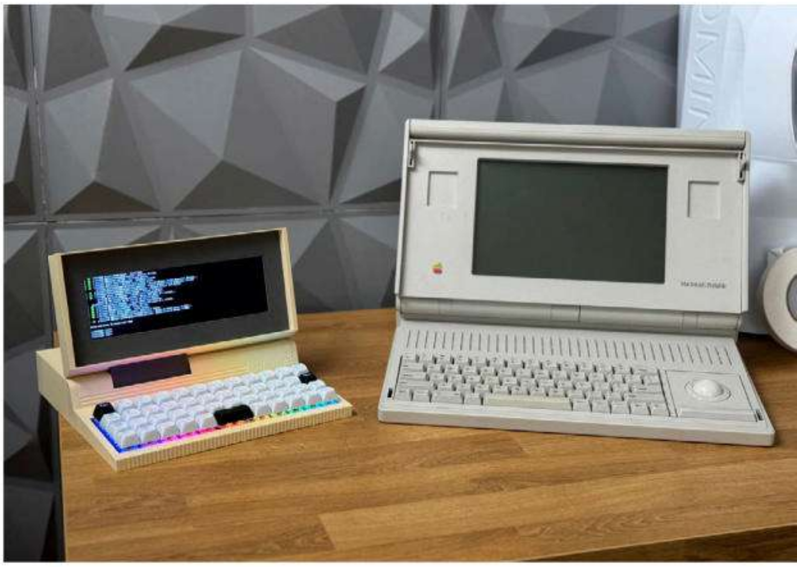
By **Ominous Industries**

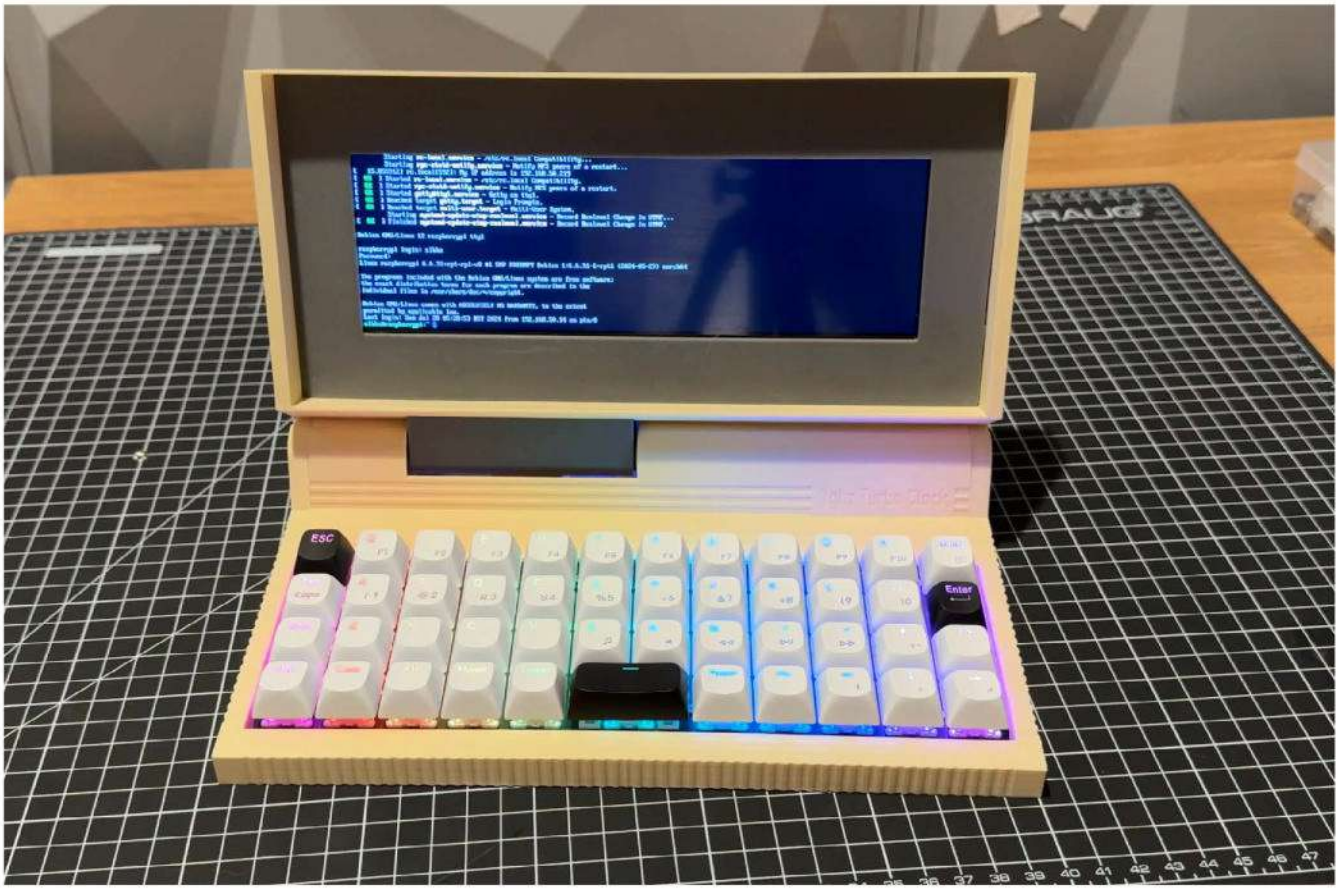
hsmag.cc/RaspberryPiZeroLaptop

There is a school of thought that says computing went wrong when we added graphical interfaces. With the command line – goes this line of reasoning – the only people who can interact with computers are those who know what they're doing. When you add a graphical interface, anyone can come along and start pointing and clicking willy-nilly, and the whole joyous human-machine interface has to be dumbed down to make it simple enough for the masses.

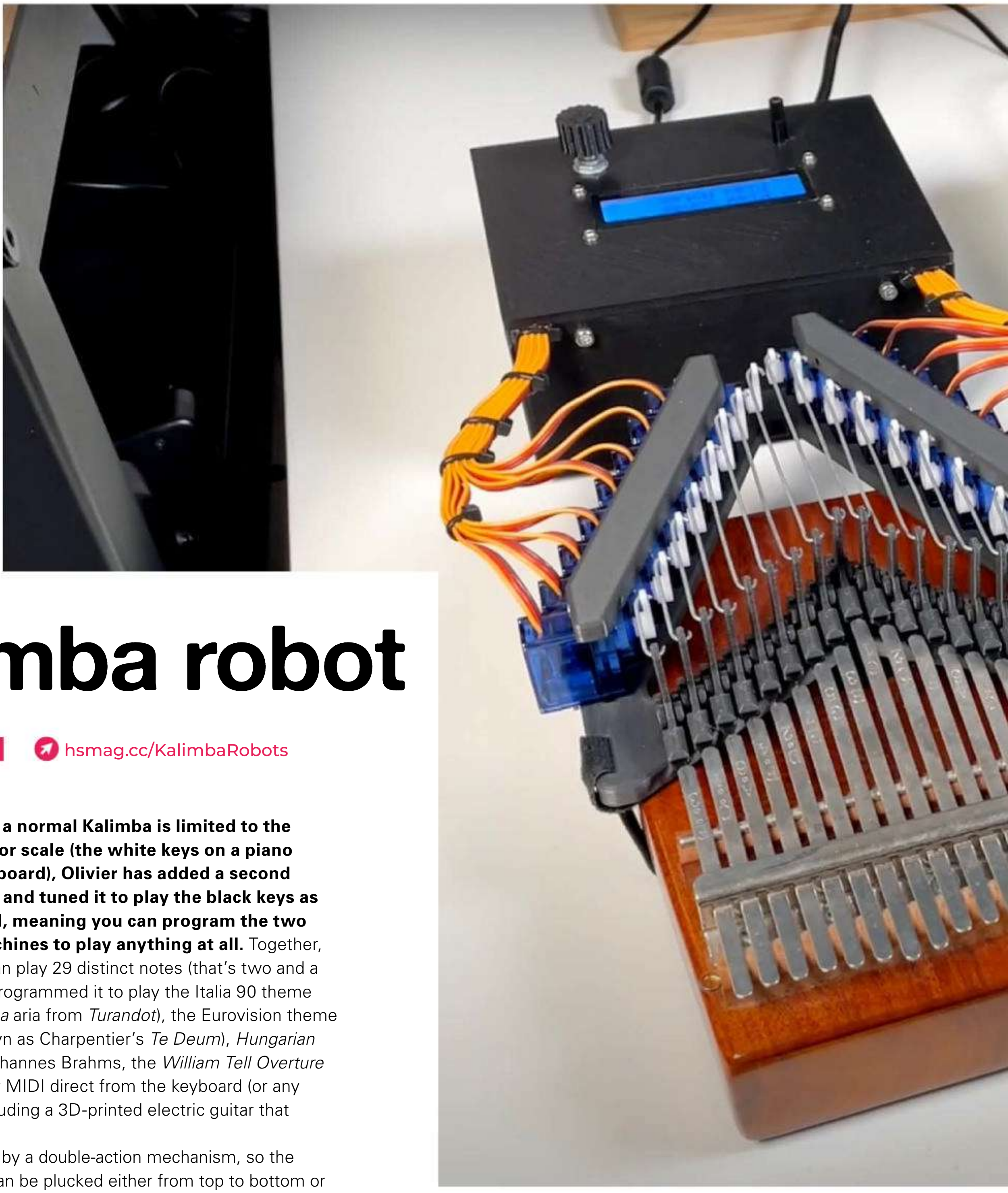
Thankfully we haven't seen anyone espouse this idea for a long time: more computer users nowadays were raised with a GUI than without, so common line natives are a shrinking segment of the population. However, there are times when we wish we could get work done without being plugged into a giant distraction machine. Wouldn't it be nice if you could have a machine that just created and edited text files, with nothing else to divert your attention? That's part of the thinking behind this Raspberry Pi Zero 2W laptop by Ominous Industries. It's been deeply inspired by the Macintosh Portable, uses a command line-only variant of the Raspberry Pi OS operating system, and in a nod toward modernity, uses a 47-key backlit RGB keyboard.

It's also a pretty simple build for anyone who fancies having a go: the STL files are all shared on **cults3D.com**, and have been designed to find on any printer with a build volume of 18 × 18 × 18 cm or larger. Electronics include the aforementioned Raspberry Pi Zero 2W, a "seven- or eight-inch" LCD screen and its driver board, and an Adafruit Power Boost 1000, which powers the screen and the Raspberry Pi. As it's command line only, there's no need to a mouse, so there's only one port – a Micro USB charging port. Jony Ive would surely approve. □





Above People have asked the maker what he's going to do with this machine, to which the answer is "I don't know, but I made, it, and I like it"



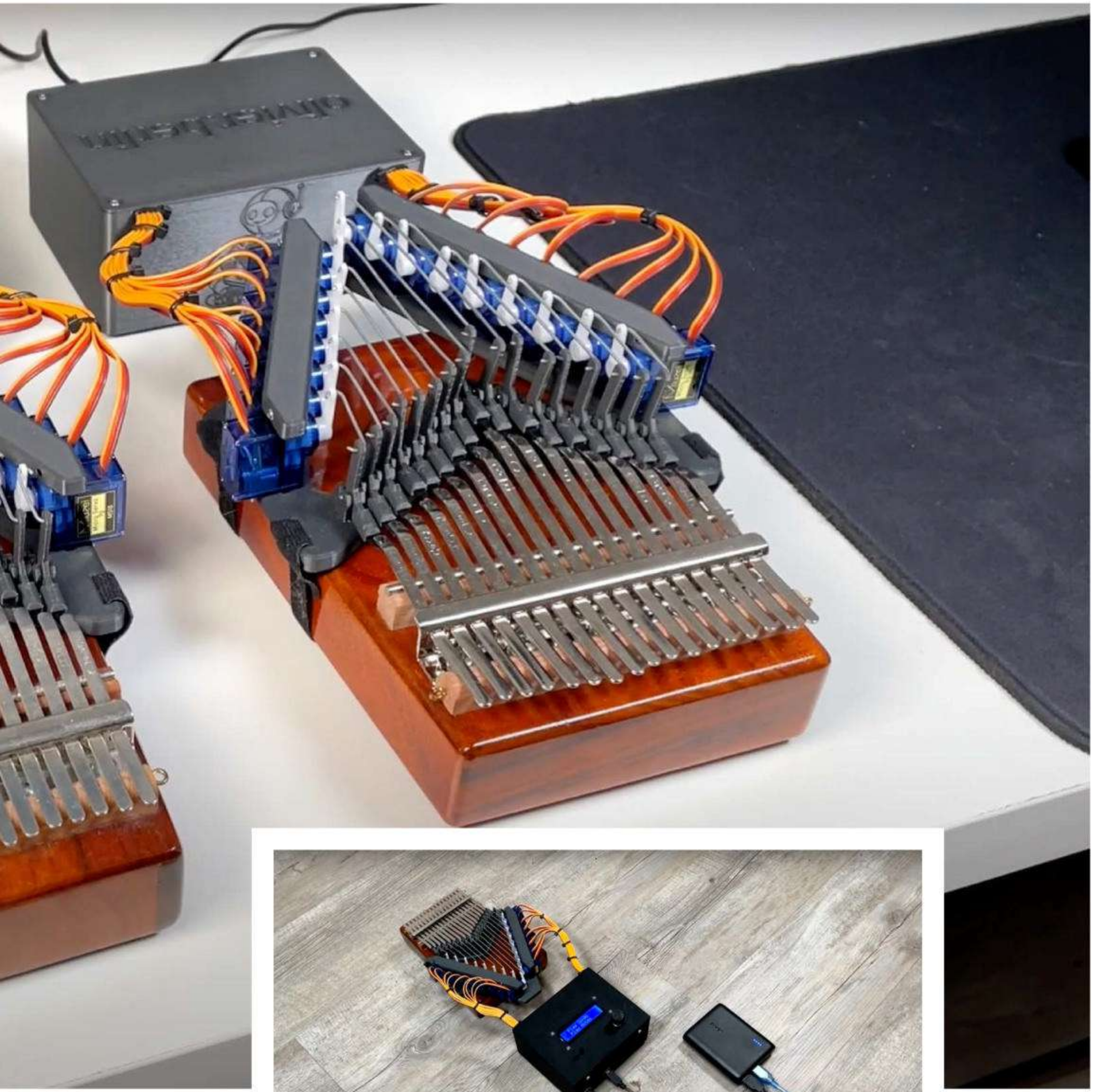
Kalimba robot

By **Olivier Berlin**

hsmag.cc/KalimbaRobots

While a normal Kalimba is limited to the major scale (the white keys on a piano keyboard), Olivier has added a second one and tuned it to play the black keys as well, meaning you can program the two machines to play anything at all. Together, these two machines can play 29 distinct notes (that's two and a bit octaves). Olivier's programmed it to play the Italia 90 theme (aka the *Nessum Dorma* aria from *Turandot*), the Eurovision theme music (otherwise known as Charpentier's *Te Deum*), *Hungarian Dance Number 5* by Johannes Brahms, the *William Tell Overture* and more. You can play MIDI direct from the keyboard (or any other MIDI device, including a 3D-printed electric guitar that Olivier has also built).

Each note is plucked by a double-action mechanism, so the tines of each kalimba can be plucked either from top to bottom or from bottom to top, and each kalimba has its own Arduino, which remembers what state (up or down) each of its mechanisms is in, so the servos don't have to reset every time they pluck a note. Our favourite feature is that it doesn't even need to be connected to a computer or keyboard – there's also an LCD screen and rotary encoder, so that the user can choose from a menu of pre-saved tunes stored on the Arduino's memory. □



Above Unbelievable for a device with so many servos, the kalimbas play output from a MIDI keyboard with next to no latency

Objet 3d'art

3D-printed artwork to bring more beauty into your life

This is the Statial.b adjustable mouse, a working prototype of a device designed to explore mouse ergonomics based on the way users grip them.

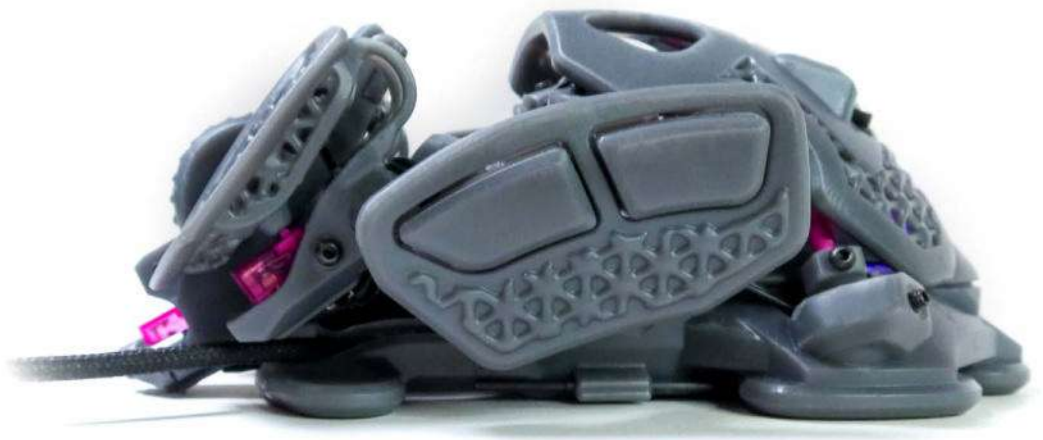
It's the second iteration of an idea that its creator, Pyott Design, first came up with over a decade ago. It's designed to be adjustable beyond anything else you can find on the market, with every plane that comes into contact with your fingers adjustable for height, pitch and roll.

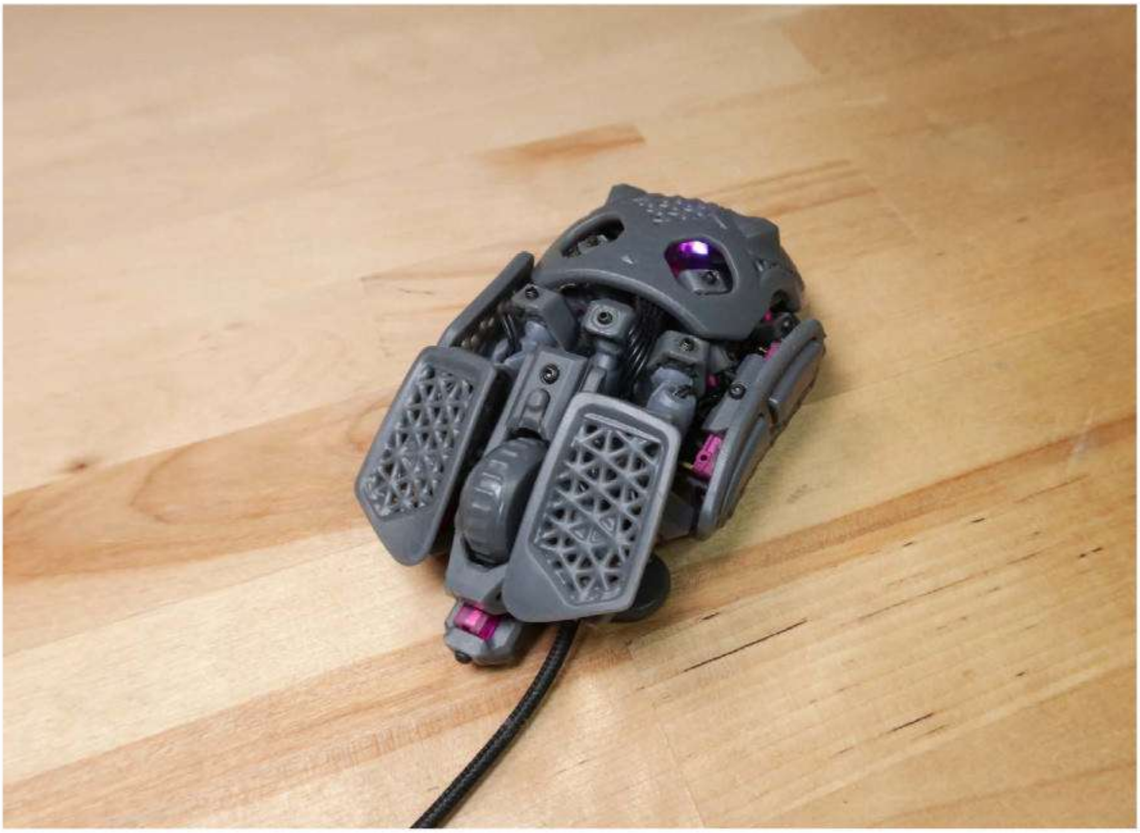
We're showcasing the intricacy of the print here, but you won't be surprised to learn that this is actually the end result of lots of trial and error, each with its own resultant improved print job. The first test print was done in clear resin, so the maker could look inside the mouse and check that the mechanical parts were moving as intended. Further testing was done in Anycubic's ABS-like Pro 2 resin, and the final print (that you see here) was done in Tough 2000 resin, on a Formlabs Form 4 resin printer.

The build is relatively complicated – you'll need a basic knowledge of 3D printing and soldering, as well as some experience of working with Arduino if you want to build this for yourself (the design files and Gerber for the custom PCB are all on GitHub).

As with anything that's super configurable, it's easy to make something that's really uncomfortable. The maker recommends setting the surfaces close to the shape of a regular mouse, then making micro adjustments until you find your favourite hand position, noting that a difference of a single degree or millimetre can feel like a lot. □

hsmag.cc/Statial-b





Right
The Tough 2000 resin allows the mouse to put up with the rigours of use

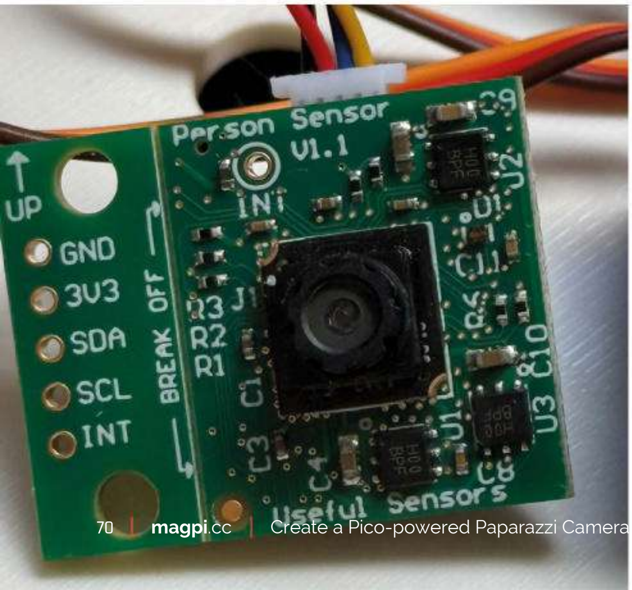
Create a Pico-powered Paparazzi Camera

Take an old point-and-shoot camera, and make it follow you around the room

Rob Miles

Rob has been playing with hardware and software since almost before there was hardware and software. **robmiles.com**.

Figure 2 The sensor is powered by a Raspberry Pi Pico fitted on the back



This project lets you create the unique experience of being hounded by the paparazzi. It uses CircuitPython code running in a Raspberry Pi Pico to control servos that make a camera notice you when you walk into the room, then track you.

This project started when the author acquired 15 broken point-and-shoot 35mm cameras which were purchased to obtain just one camera which was part of the package. However, once all the cameras had been received, tested and found to be properly broken thoughts turned to what to do with them. The original plan was to use nine of them to create an installation in which all the cameras followed you around the room. However, it was thought best to start with just one, and scale things up once permission has been obtained to do something interesting with the living room wall. You can find all the resources for the project at **hsmag.cc/picopapcam**

FIND YOURSELF

The first problem to be solved was finding the person for the cameras to move towards. The Person Sensor from Useful Sensors was built specifically for this purpose. It contains custom software connected to a small camera.

Figure 2 shows the Person Sensor. You could use this sensor to make a device that wakes up when a user appears in front of them. It detects multiple people in the frame, although the paparazzi camera will only track one.

The Person Sensor uses an I2C bus connection to deliver position data when someone is detected in the field of view of the camera. This data is received by a CircuitPython program running in a Raspberry Pi Pico to drive servos to move the camera.

I2C (or IIC) is an abbreviation for Inter Integrated Circuit, a component connection bus standard developed by Philips in the 1980s for use in colour televisions. When a CircuitPython program wants to interact with a device using I2C it creates an object to represents the hardware connection.

```
$ i2c = busio.I2C(scl=board.GP5, sda=board.GP4)
```

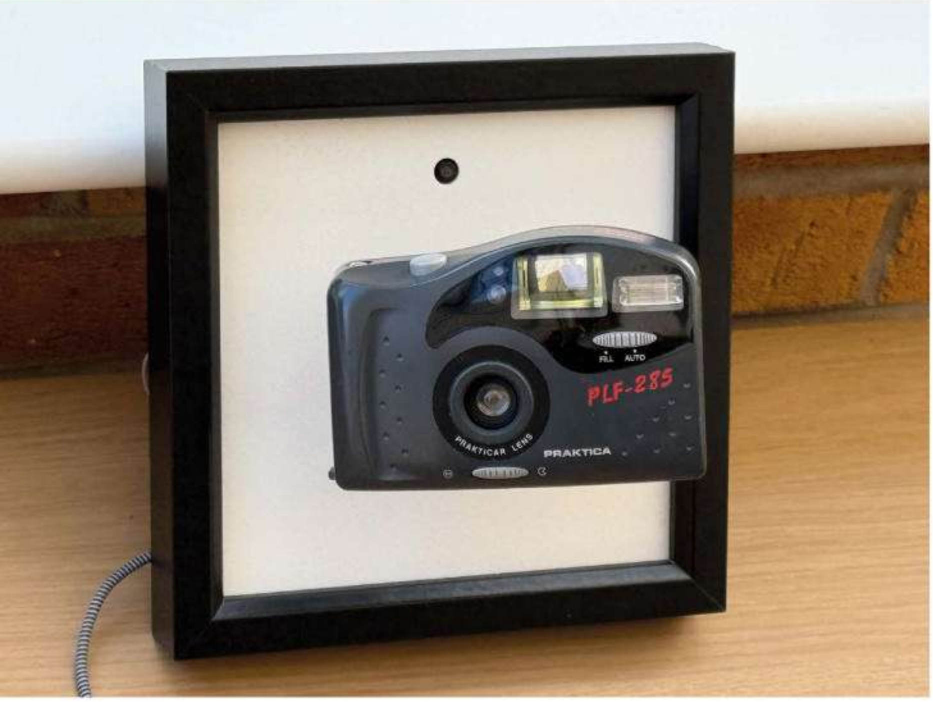
The I2C variable created by the code is connected to the specified GPIO (General Purpose Input/Output) pins on the Pico. The scl signal provides the clock, and the sda the data.

A TOUCH OF CLASS

The Paparazzi Camera program has been designed using Python classes. The program contains a class called `PersonSensor` which uses the I2C connection to get position values out of the sensor. When we create the `PersonSensor` instance we pass it the I2C interface it is going to use:

```
$ personSensor = PersonSensor(i2c)
```

The statement above creates an instance of the `PersonSensor` class which is called `personSensor`. The author is observing the convention that the name of a



The first gives the address of the person sensor on the I2C bus, the second gives the size in bytes of the data block the sensor sends. Now we can create the data block to store the information we want from the sensor.

class (**PersonSensor**) starts with an upper-case letter, but the name of a variable (the instance **personSensor**) starts with a lower-case letter. Let's have a look at the **PersonSensor** class.

```
$ class PersonSensor():
    def __init__(self, i2c):
        self.i2c = i2c
```

The code above shows the first part of the **PersonSensor** class. The **__init__** method is called when a **PersonSensor** instance is created and is passed the I2C connection which will be used to communicate with the sensor. This value is stored inside the **PersonSensor** for use when a position is requested. The **PersonSensor** class contains a **get_faces** method which will get any faces that have been detected.

```
$ faces = personSensor.get_faces()
```

The **get_faces** method returns a list of face information from the sensor. Let's look at how it does this, starting with a few important numbers.

```
$ PERSON_SENSOR_I2C_ADDRESS = 0x62
PERSON_SENSOR_RESULT_BYTE_COUNT = 40
```

The code above creates two named values.

```
$ get_faces(self):     read_data =
bytearray(PERSON_SENSOR_RESULT_BYTE_COUNT)
```

The code above shows the start of the **get_faces** method. It creates a byte array called **read_data** to hold the incoming information. A byte is a value represented by eight binary bits. Most computers store data in byte-sized locations.

Now that we have somewhere to put the data we can fetch it from the sensor. But before **get_faces** does this, it must make sure that it has exclusive access to the I2C connection. This ensures messages are not corrupted by other processes using the sensor at the same time.

```
$    while not self.i2c.try_lock():
        pass
```

One process at a time can establish a lock on the I2C bus object. The **while** loop above repeatedly tries to get hold of the lock by calling the **try_lock** method. When I2C is available the **try_lock** method returns True and **get_faces** continues.

```
$    self.i2c.readfrom_into(PERSON_SENSOR_I2C_
ADDRESS, read_data)
    self.i2c.unlock()
```

The code above shows the part of **get_faces** which →

QUICK TIP

The project hardware has the capability to control 16 servos, so it would be possible to extend the project to add extra cameras.

Figure 1
The author wanted the camera to seem to float out of the picture frame behind it. This turned out to be irritatingly hard to realise

QUICK TIP

The person sensor can also be trained to perform some facial recognition, so you could create a paparazzi camera which only follows you.

YOU'LL NEED

- An old "point-and-shoot" film camera
- A Raspberry Pi Pico
- A PCA9685 16 Channel 12-bit PWM Servo motor driver I2C.
- Four SG90 servo motors
- A Useful Sensors person detector
- A six-inch picture frame
- Some 3D-printed support parts
- Wire to connect the servos to the camera

QUICK TIP

Several devices can share a single I2C connection. In this project the Person Sensor and the Servo Controller use the same one.

reads from the person sensor. When we have finished using the I2C bus it is very important to unlock it, so that it can be used by other parts of the program.

The `readfrom_into` function fetches a block of data from the bus and stores the values in an array of bytes called `read_data`. We now have a block of data which the Person Sensor has sent us. A software ninja would be able to use the byte values directly from the `read_data` array, but the rest of us would rather like to unpack the array and give sensible names to the things in it.

PACKET UNPACKING

Python provides a very powerful struct mechanism for working with low-level data. We can create descriptions of a data format in the form of strings which are then used to unpack binary data. As an example, the string "BBH" would tell struct to decode two unsigned signed eight-bit values followed by an unsigned 16-bit value held in two bytes. The code in the Paparazzi application uses struct-powered code to determine how many faces were detected and then creates a Python dictionary for each of the faces which can be used to determine where the face is in the frame viewed by the camera.

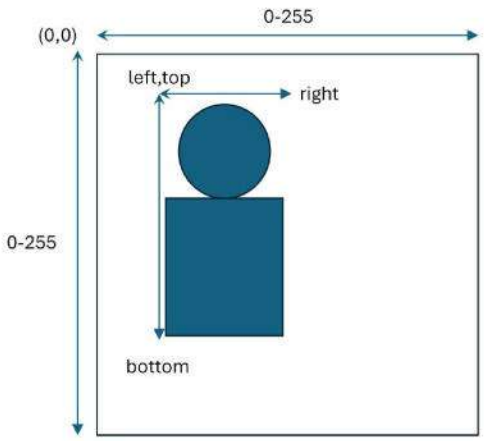


Figure 3 Note that the origin for the figure is in the top left hand corner

TAKING AIM

Now that we have the person location data from the sensor the next thing to do is convert this into information that can be used to control the servos and aim the camera. The person data returned from the sensor takes the form of a Python dictionary containing a set of keys. The values for each key have been extracted from the binary data received from the Person Sensor.

```
$  {'box_bottom': 161,     'id': -1,     'is_
facing': 0,    'box_confidence': 69,    'box_
left': 52,    'box_top': 12,     'id_confidence':
0,     'box_right': 170    }
```

Above is a typical reading. The box left, right, top and bottom values give the position in the frame of a box containing the person identified.

QUICK TIP

If a naughty process fails to unlock the I2C connection when it has finished this would cause the whole device to get stuck.

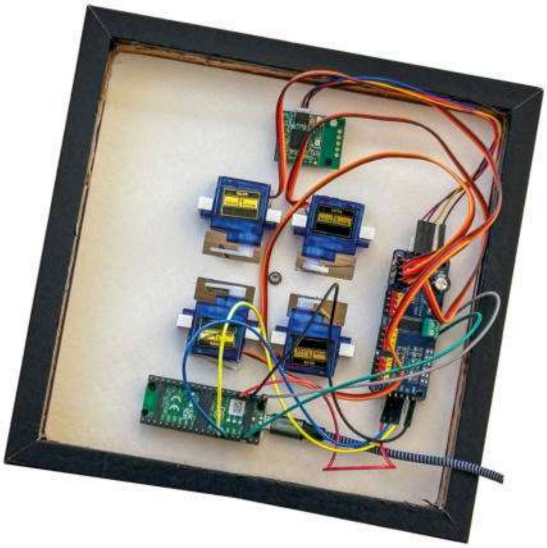


Figure 4 The servo controller is on the right of the board

Figure 3 shows how the data appears. The Paparazzi Camera needs the X and Y coordinates of the position of the centre of this box, so we need some code to use these values and work out the position:

```
$ x = face["box_left"] + (face["box_right"]
-face["box_left"])/2.0
$ y = face["box_top"] + (face["box_bottom"] -
face["box_top"])/2.0
```

The two statements above set x and y to the position of the centre of the box. This gives us position values in the range 0-155. We can use these values to tell the servos where to point the camera.

SERVO SERVER

The Paparazzi camera uses servos to move the camera around. Servos deliver precise rotational movement, usually in the range 0 to 180 degrees. The servo output position is set using a pulse-width modulated (PWM) control signal. In PWM the control signal switches rapidly between on and off. A program can control the amount of power delivered by varying the length of the "on" time. The Raspberry Pi Pico can produce PWM signals, but for this project it was decided to use a PCA9685 controller board. This makes it easier to connect the servos and distribute power to them. The Paparazzi camera program contains a class called `ServoManager` to manage the servos:

```
$ servoManager = ServoManager(i2c)
$ servoManager.set_positions(90,90,90,90)
```

The first statement creates a `ServoManager` instance, and the second statement sets the position of each servo to its mid-point. The present version of the software sets the servos to different positions depending on the position of the person in the frame.

```
$ servoManager.set_positions(30,140,30,140)
```

The above statement moves the two left-hand servos in and the two right-hand servos out. The servos are connected to wires which would pull the camera to the left. Other directions are obtained by different combinations of these signals. Later versions will perform more precise targeting.

Figure 4 shows the components arranged in the back of a picture frame. The servos are fitted with levers connected to wires which pull on the camera when the servos are turned.

Creating the electronics and the software turned out to be the easiest part of this project. Suspending a camera in front of a picture frame and then rotating it was much harder. It is crucial that the camera is supported as closely as possible to its centre of gravity, otherwise the servos must support the camera weight as it pivots away from them.

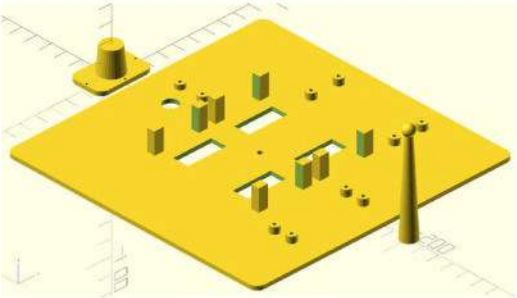


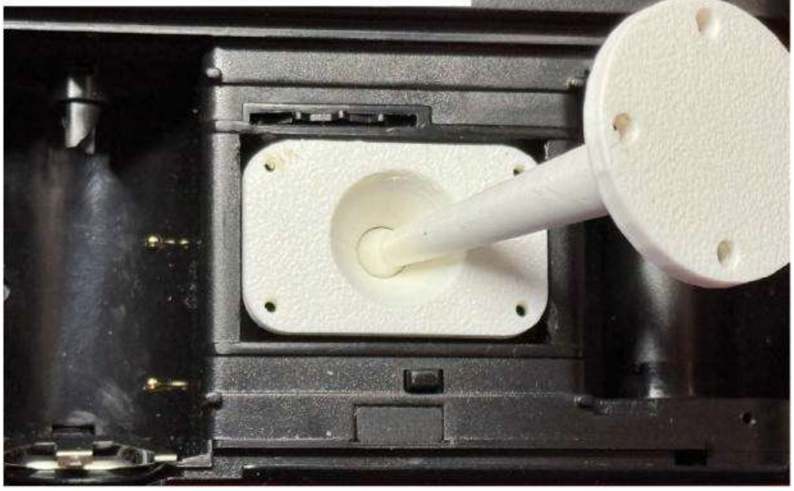
Figure 5 The design was created using OpenSCAD

BUILDING THE DREAM

Figure 5 shows the 3D-printed parts used in the project. The camera support post is bolted into the backplate and the camera support snaps on top of it. The camera support is glued into the camera.

Figure 6 shows the camera support plate inside the back of the camera and the camera post fitted into it. The top of the pillar snaps into a hole in the support plate, which is positioned behind the camera lens, as close to the centre of the camera as possible. The holes in the corners of the support plate are for wires which are connected to the servo arms. This solution works quite well, but there is a problem with the camera twisting on the ball joint because the camera is not perfectly balanced horizontally. The author is presently experimenting with universal joints to attach the camera to the support pole. This would allow for pan and tilt, but not rotation.

Figure 6 It turns out to be surprisingly easy to create 3D-printed components that snap together



> "The Person Sensor detects multiple people in the frame, although the Paparazzi Camera will only track one"

Figure 7 shows test control wires connected between the camera support plate and the servos. The length of each wire had to be adjusted to make the camera balance properly.

HAPPY ENDINGS?

The author is mostly happy with the finished system, although there are improvements to be made. The camera support needs to be refined so that it is easier to balance the camera, and the software could be improved so that the camera can point at any location, not just particular points. He is still waiting for permission to make the other eight paparazzi. □

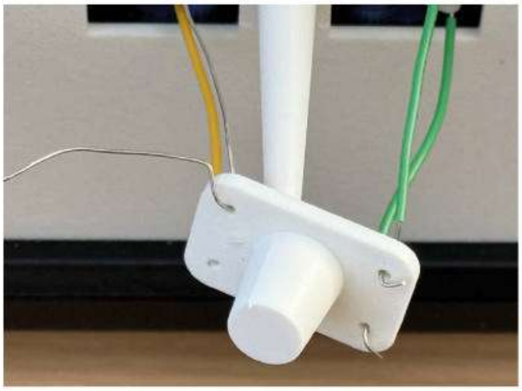
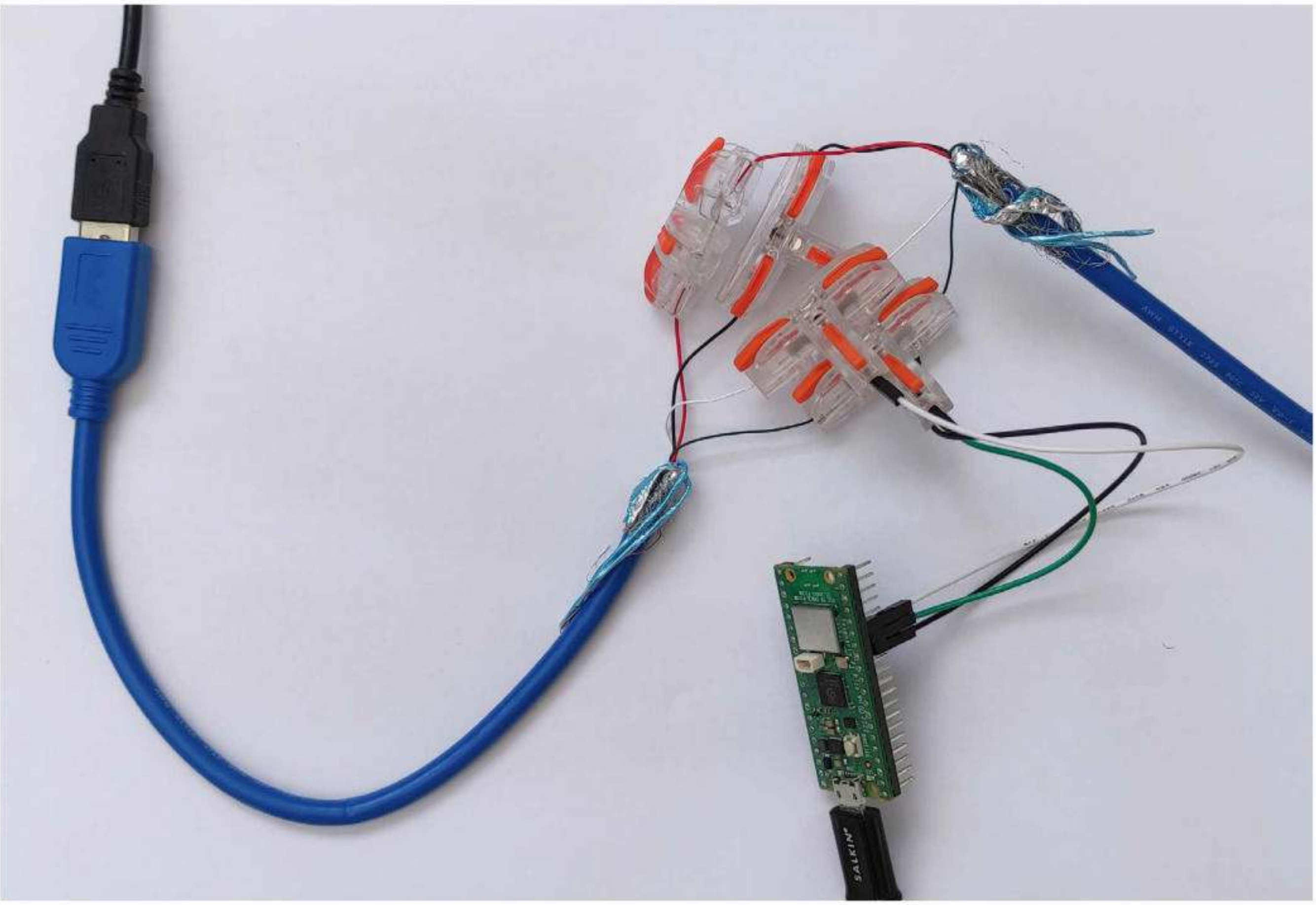


Figure 7 It would have been possible to use just two servos. The design uses four to increase the power available

QUICK TIP

When a programmer creates a software driver for an I2C device they study the datasheet for the device, determine the format of the data that it uses and then create struct descriptions for the packets of data. In other words, you now know how device drivers are made.



Make a Pico USB sniffer

Create an inexpensive USB data packet sniffer based on a Raspberry Pi Pico

Phil King

A long-time Raspberry Pi user and tinkerer, Phil is a freelance writer and editor with a focus on technology.

A **USB sniffer is a tool that enables you to monitor the packets of data being sent and received via a USB port with a device attached to it.** The captured data can then be analysed – very useful for developers during the testing of hardware, drivers, and applications.

While software-based sniffer tools such as Wireshark are available, there are several reasons why you may want to use a hardware-based USB sniffer instead. One of the key advantages is the ability to monitor USB traffic on another host computer, or even a device such as a game console, not just the PC doing the analysing.

The downside is that hardware sniffers tend to be fairly expensive, some costing £100 or more. Thanks to Alex Taradov's USB Sniffer Lite project, however, you can now build your own based on a Raspberry Pi Pico and a chopped up USB cable. We'll guide you through the process of wiring it up to intercept the traffic between a USB device and host computer, as well as explaining how to flash the firmware to Pico and use it via a serial interface. Let's get sniffing…

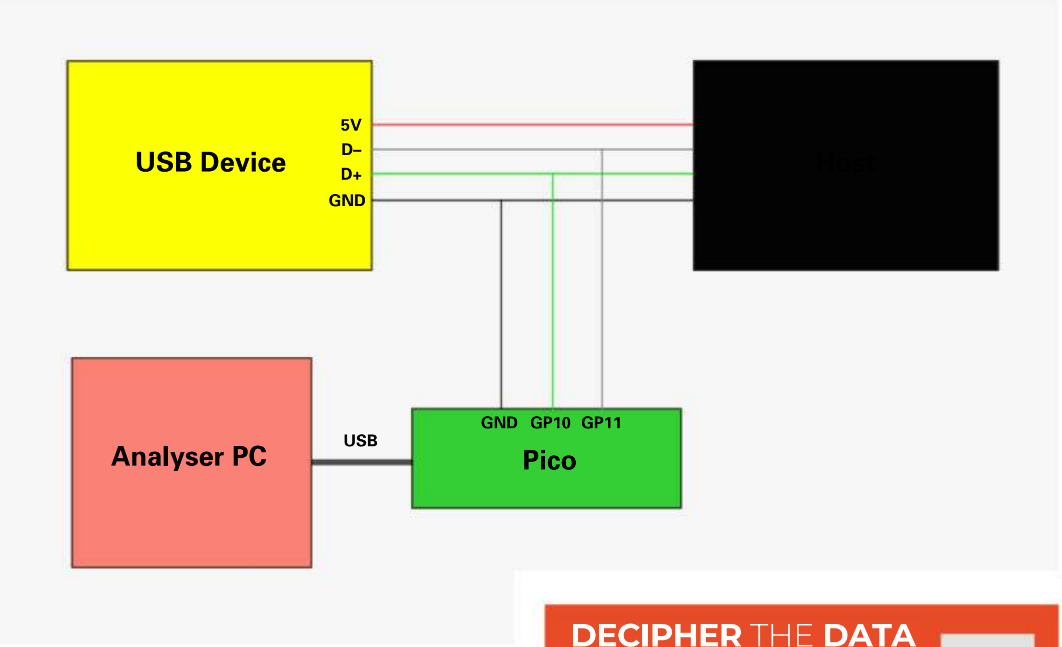


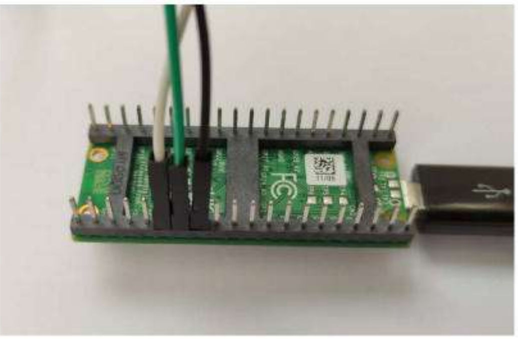
Figure 1 The wiring diagram for connecting the spliced USB cable to a device, host, and Pico

Left It may look like a rat's nest with all the wiring, but this Pico-based USB sniffer is inexpensive and really works

FLASH THE FIRMWARE

First of all, we'll flash Alex Taradov's USB Sniffer Lite firmware to Pico and check that it's working OK via a serial interface in the terminal on the connected PC.

Head on over to the project's GitHub repository (**magpi.cc/usbsniffer**). In the **bin** folder, you'll find a firmware file called **UsbSnifferLite.uf2**. Click on it, then hit the download arrow to save it.



Above Three pins are used on Pico. We used jumper wires to connect them to our spliced USB cable

While the GitHub readme notes state that "the compiled binary needs to have CRC value updated to be recognized by the RP2040," this only applies if you compile your own from the source code in the repo. So don't worry about that – the UF2 file you just downloaded is ready to use.

DECIPHER THE DATA

Befuddled by the data captured? Here are some of the terms you may encounter…

- LS / FS: Low / Full speed.
- DATA0 / DATA1: These mark the start of an odd or even data packet.
- DATA2, MDATA: A high-speed data packet (USB 2.0).
- ACK / NAK: Data packets accepted / not accepted.
- NYET: Data not ready yet (USB 2.0).
- STALL: Transfer impossible.
- SOF: Marks the start of a frame.
- IN / OUT: Address for device-to-host transfer or vice versa.
- SETUP: Address for host-to-device control transfer.

For more information, see the Wikipedia page on USB communications: hsmag.cc/USBComms.

To flash the firmware to Pico, it's the usual process. Hold down Pico's BOOT button while connecting it to your computer via USB, then release the button and it'll be mounted as a storage drive. Drag the UF2 to its folder to copy it across; it'll then unmount automatically.

If it's working correctly, it'll create a virtual COM-port (VCP) that can be accessed via the command-line interface from any computer. Open up a terminal window and enter the following command to search for the device:

```
ls /dev/tty.*
```

This will show a list of all available serial ports. The Pico USB sniffer will show up with a name starting with 'tty.usbmodem' followed by a hex number. For instance, ours was 'tty.usbmodem6DFE8E421'. →

YOU'LL NEED

- **Raspberry Pico / Pico W / Pico 2**
- **USB cable** female to male
- **Wire stripper/ cutter tool**
- **Wire connectors** (optional)
- **Jumper wires** (optional)

To connect to it, you can use the screen command. In our case, that's:

```
screen /dev/tty.usbmodem6DFE8E421
```

Change the hex number if yours is different.

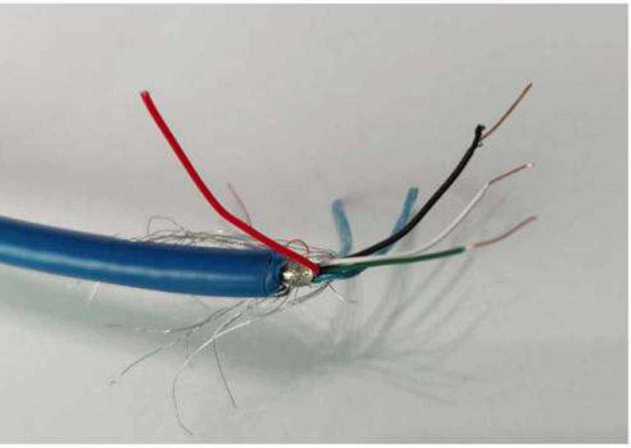
The USB Sniffer Lite's interface should then appear, offering a range of options – we'll go through these later.

Now you know it's working, you can exit the serial connection: press **CTRL** + **A** + ****, then answer **Y** when prompted with "Really quit and kill all your windows?"

WIRING IT UP

To intercept traffic between a USB device and the computer – whether it's the one you're connecting to Pico, or a different one – you'll need to splice the connecting USB cable to tap into it. To make it easy to switch between different USB devices, we're using a female-to-male USB cable.

Using your wire stripper/cutter tool, cut the cable in half. Then, on one half, use an appropriately sized stripper hole to strip off a length – 5cm or more – of the outer insulation. You may find that under this, there is a layer of shielding and coaxial braid, along with a bare copper wire – pull all of this back to access the inner wires.



Above You need to strip the cable's outside insulation, then the individual wires – we still need to do the red one here

QUICK TIP

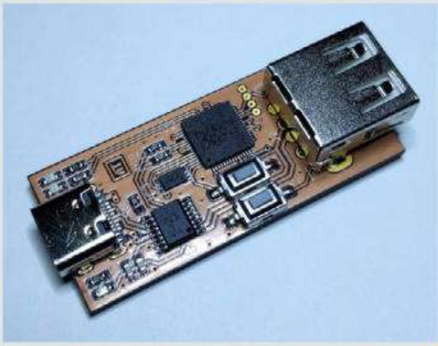
Strip enough insulation from the main cable so that the individual wires are long enough to make all the connections easily.

Of these, you only need the red, green, white, and black wires. So push the other two back out of the way. Using a smaller hole on the wire stripper tool, strip 1cm or so off the end of each of the four wires to reveal the copper wire.

Repeat the whole process for the other cut end of the USB cable. You should now have four stripped wires (red, green, white, and black) on each end, ready to connect. You'll also need some more wire to also connect three of these (all bar the red 5V power wires) to Pico's pins – we opted to use standard male-to-female jumper wires.

DEDICATED BOARD

Don't like all that messy wiring? Alex Taradov has also designed a dedicated hardware board to use instead, based around Pico's RP2040 chip. The schematics and Gerber files can be found in the hardware folder of the project's GitHub repo, along with STL files for a 3D-printed case.



Right Alex Taradov has also designed this RP2040-based board that does the same job

To make the three-way connections (and two-way power), you could use wire-wrapping, solder, or standard electrical block terminals, but we found some nifty three-way snap-fit connectors on Amazon (**hsmag.cc/3WayConnect**) that were easy to use and made for a neater finish.

You need to connect all the wires up as in **Figure 1**. The three wires going to Pico are as follows:

Pico Pin	Function	USB Wire Colour
GND	Ground	Black
GPIO 10	D+	Green
GPIO 11	D-	White

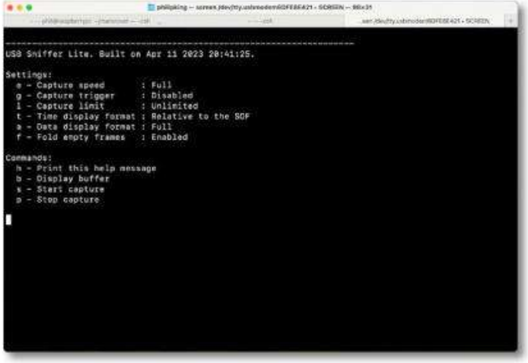
You can now connect any device to the female end of the USB cable, and plug the male end into any computer (or other host device). The latter could even be plugged into another USB port on the same computer you're using to analyse the data with Pico.

CAPTURING DATA

With everything wired up correctly, you can now start capturing some USB data to analyse. As before, connect Raspberry Pi Pico to a computer and open a serial connection to it with the screen tool in a terminal.

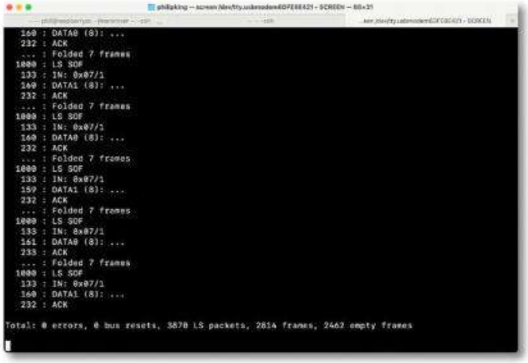
Right
Open a serial connection to the Pico USB sniffer and you should see this menu

Below
Data captured from the USB sniffer and displayed in the terminal



philipking — screen /dev/tty.usbmodem6DFE8E421 · SCREEN — 80×31

```
-------------------------------------------------------------------------
USB Sniffer Lite. Built on Apr 11 2023 20:41:25.

Settings:
  e - Capture speed       : Full
  g - Capture trigger     : Disabled
  l - Capture limit       : Unlimited
  t - Time display format : Relative to the SOF
  a - Data display format : Full
  f - Fold empty frames   : Enabled

Commands:
  h - Print this help message
  b - Display buffer
  s - Start capture
  p - Stop capture

```



philipking — screen /dev/tty.usbmodem6DFE8E421 · SCREEN — 80×31

```
  160 : DATA0 (8): ...
  232 : ACK
  ... : Folded 7 frames
 1000 : LS SOF
  133 : IN: 0x07/1
  160 : DATA1 (8): ...
  232 : ACK
  ... : Folded 7 frames
 1000 : LS SOF
  133 : IN: 0x07/1
  160 : DATA0 (8): ...
  232 : ACK
  ... : Folded 7 frames
 1000 : LS SOF
  133 : IN: 0x07/1
  159 : DATA1 (8): ...
  232 : ACK
  ... : Folded 7 frames
 1000 : LS SOF
  133 : IN: 0x07/1
  161 : DATA0 (8): ...
  233 : ACK
  ... : Folded 7 frames
 1000 : LS SOF
  133 : IN: 0x07/1
  160 : DATA1 (8): ...
  232 : ACK

Total: 0 errors, 0 bus resets, 3870 LS packets, 2814 frames, 2462 empty frames
```

The USB Sniffer Lite menu offers a range of options, which we'll take a look at in a moment. First, let's try a quick data capture. Press the **S** key to start the capture, wait a few seconds, then press **P** to stop the capture. You should then see a list of the data that was captured – this data buffer can also be displayed again at any time by pressing **B**. Pressing **H** will bring up a help message with all the settings, as in the original menu…

- Capture speed (**E**) – this toggles the speed between Low and Full.
- Capture trigger (**G**) – enable / disable. When enabled, the capture will pause until the trigger pin (GPIO18) is pulled low, such as with a connected switch.
- Capture limit (**L**) – set to 100/200/500/1000/2000/5000/10000 packets or unlimited.
- Time display format (**T**) – relative to the first packet / previous packet / SOF / bus reset.
- Data display format (**A**) – full / limit to 16 bytes / Limit to 64 bytes / Do not display data.
- Fold empty frames (**F**) – enable / disable. When enabled, this shortens the displayed information by combining consecutive empty frames (those with no packets other than IN/NAK) into one entry.

We plugged a USB game controller into the female USB socket of the cable and inserted the male end into a second USB port on our computer already connected to Pico. We then captured some data manually, by pressing **S** and then, after a while, **P**. To put the data in a file, you can just copy and paste it from the terminal. We also tried capturing data between the controller and a Nintendo Switch console.

While we don't have the scope in this tutorial to explain in detail how to analyse your captured data, see the 'Decipher the Data' boxout for some pointers.

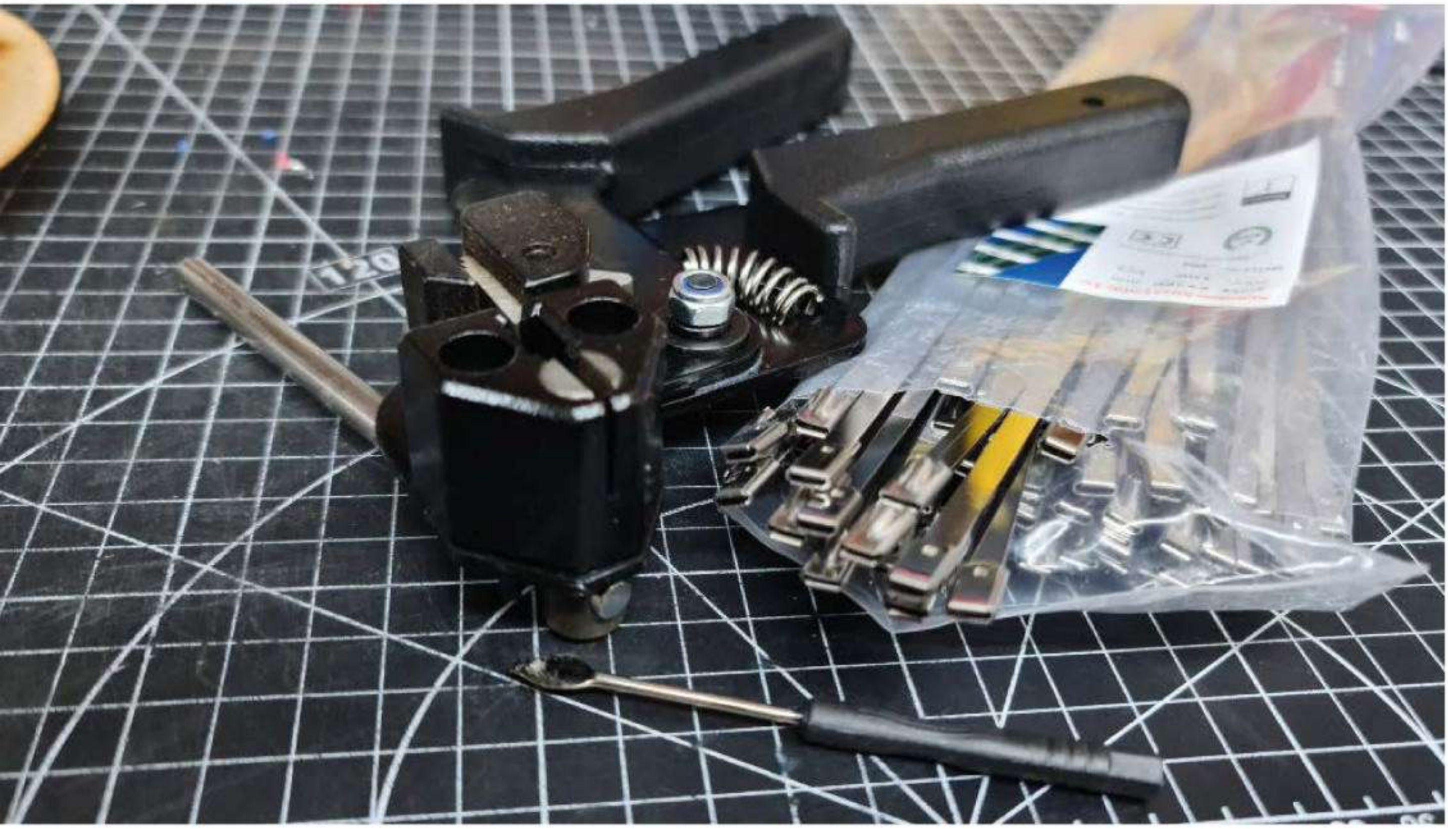
Now you've learned how to turn a humble Raspberry Pi Pico into a hardware USB packet sniffer, you can capture and analyse the data going between any two devices plugged into your setup. What will you use yours for? Do let us know. □

QUICK TIP

You can sniff packets between a device and a different host, or the one connected to Pico for – just use a different USB port.

> "The captured data can then be analysed – very useful for developers"

Above
You'll need a wire stripper tool to cut the USB cable and strip the wires on each end of the splice



Needful things: stainless steel tie tensioning tool

Find a hidden hero for tense and heated situations

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

There's an old joke that modern civilisation will collapse if a virus evolves that can eat duct tape and cable ties. With a stainless steel tie tensioning tool, you can at least partially forestall the end of days and add a new mechanical fixing method to your creative arsenal.

Cable ties are one of the best terrible ideas of the modern age. They're easy to use, low cost, and relatively low tech. Wrap a cable tie around the things you want bundled together, pull the tie tight, and cut off the excess – it's as easy as that. Sadly, cable ties are also one more source of waste plastic for the environment and they don't always do well in situations where heat, heavy loading, or ultraviolet light are a factor. The often overlooked stainless steel cable tie can offer a much more permanent, recyclable alternative, but they seldom get the opportunity to show their worth. This is partly because they're not as cheap or ubiquitous as the plastic cable ties, but also because they require a little bit more force to tighten up than their flimsier plastic relative.

Far Left
You can get a steel tie tensioner and a good selection of stainless ties for less than £15/$20. It's surprising how useful they can be if you keep them in your home or vehicle toolkit

Below
Metal ties usually use a ball bearing ratchet mechanism to lock the tie in place. You can remove the tie and reuse it by poking a screwdriver into the exposed end of the ratchet mechanism to dislodge the ball. Some tensioning tools come with a dedicated tool to do this

Unlike a jubilee clip, which has a built in worm drive that pulls the tie tight, the stainless steel tie relies on the pulling force of the user and a ratchet or wedging mechanism to cinch it up tight. For one or two ties this isn't a big problem, and you can get away with using a pair of pliers and a lever to tighten the tie. On jobs where you're going to use more than one or two stainless ties, a tie tensioning tool is pretty much essential for your sanity and for the integrity of your fingers.

A tie tensioning tool (also known as a tie tool or tie gun) takes the pain out of tightening and trimming up cable ties by regulating the force you apply to the tie, and optionally trimming away any excess metal. Tie tools exist for both plastic and metal cable ties, but the metal version is the one that provides the most benefits to the user. The stainless steel tie tensioning tool looks something like a cross between a cable stripper and a ray-gun, with a spring loaded handle, a beak-like grip that pulls on the cable tie, and a sturdy looking metal lever at dangles out of the back of the tool.

> "A tie tensioning tool (also known as a tie tool or tie gun) takes the pain out of tightening and trimming up cable ties"

To use the tie tensioner, wrap a metal tie around your chosen objects and pass the end of the tie through the ratcheting head and take up the slack. Place the end of the tie into the tie tensioning tool, and squeeze the handle to cinch up the tie. Each squeeze of the handle will pull more of the tie through the grip, increasing the tension. Once you have reached the appropriate tension, pull the large lever at the back of the tool to engage the guillotine blade and snip the excess metal from the tie.

If you find yourself regularly working with higher-temperature equipment such as car engines, furnaces, or 3D printers, having access to stainless steel ties and a tensioning tool can make a lot of sense. It's a very inexpensive purchase, and whether you need to temporarily fix a falling exhaust or run silicone tubes through a heated chamber, the tensioning tool turns an awkward job into an easy one, and lets you to get the job done without causing strain to your hands. □

QUICK TIP

Although a metal tie isn't razor sharp, it is possible to damage yourself or equipment if you overtighten a tie.

DIY CNC Lathe and custom G-codes

Being able to write G-codes enables all kind of custom machines, so let's look at a simple small CNC conversion

Jo Hinchliffe

With a house and shed full of lathes, milling machines, 3D printers and more, Jo (AKA Concretedog) is a constant tinkerer and is passionate about making. Obsessed with rockets and robots and much more besides, he often releases designs and projects as open source.

A CNC lathe is something I've always wanted. I've operated manual lathes a fair amount, and I've used CNC mills and routers, but I've always pondered how CNC lathe operations work. As I had no project in mind for it I didn't want to spend a fortune on a pre-built CNC lathe, so I set out to convert one.

The lathe in question is a small hobby lathe called a Unimat 3, seen in **Figure 1**. Unimat, part of the larger EMCO company, produces lots of small lathes aimed at the hobby market. While they have limitations, they are capable of producing decent results on small parts and as such are quite desirable. I'd seen a few people convert an even smaller Unimat model, the Unimat SL, to CNC and I felt the slightly more solid Unimat 3 was a good candidate for partial conversion. Why partial? Well there are numerous levels of CNC lathe, and this project is in some ways the minimal viable conversion to allow me to play with CNC lathe techniques. I've added computer control to the movement of the cutting tool by adding stepper motors to the lead screws along the ways and the cross slide. A more complete CNC lathe conversion would have the spindle and the chuck under CNC control to co-ordinate the movements of the tools with turning of the spindle, this allow the creations of threaded parts for example. It's not beyond the realms of possibility to do this, adding some form of sensor to detect spindle movement and a computer controlled drive system, but for this stage of the project just getting the tool end of the system moving offers plenty of learning.

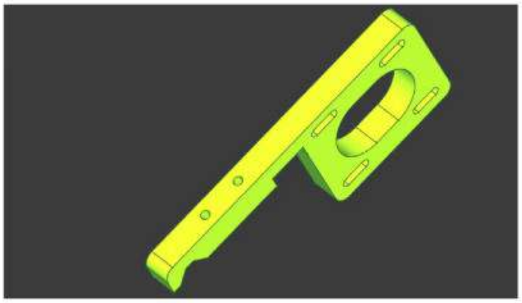


Figure 2
The adapter plate for the main ways or Z axis

With lathes of all sizes rigidity is key in terms of creating accurate work efficiently. For its size the little Unimat 3 is reasonably rigid and in early thoughts I considered adding metal parts to mount the stepper motors to the lathe to keep everything rigid. Whilst this is not a bad idea at all I realised that the actual drive system and coupling doesn't have to be that rigid at all, merely strong enough to turn the lead screws. I decided to design some non destructive add on stepper motor mounts for the Unimat 3 in FreeCAD and then 3D print them: **Figure 2**. As this is an experiment, I didn't particularly want to get into drilling holes into the lathe, and it's nice that I could totally revert the lathe to stock.

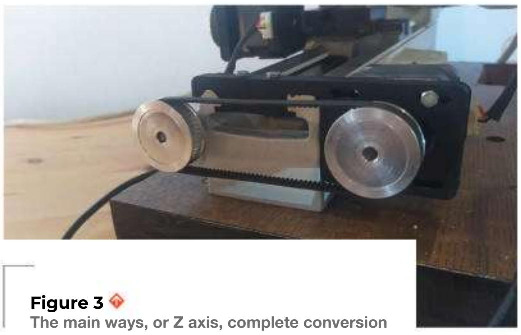
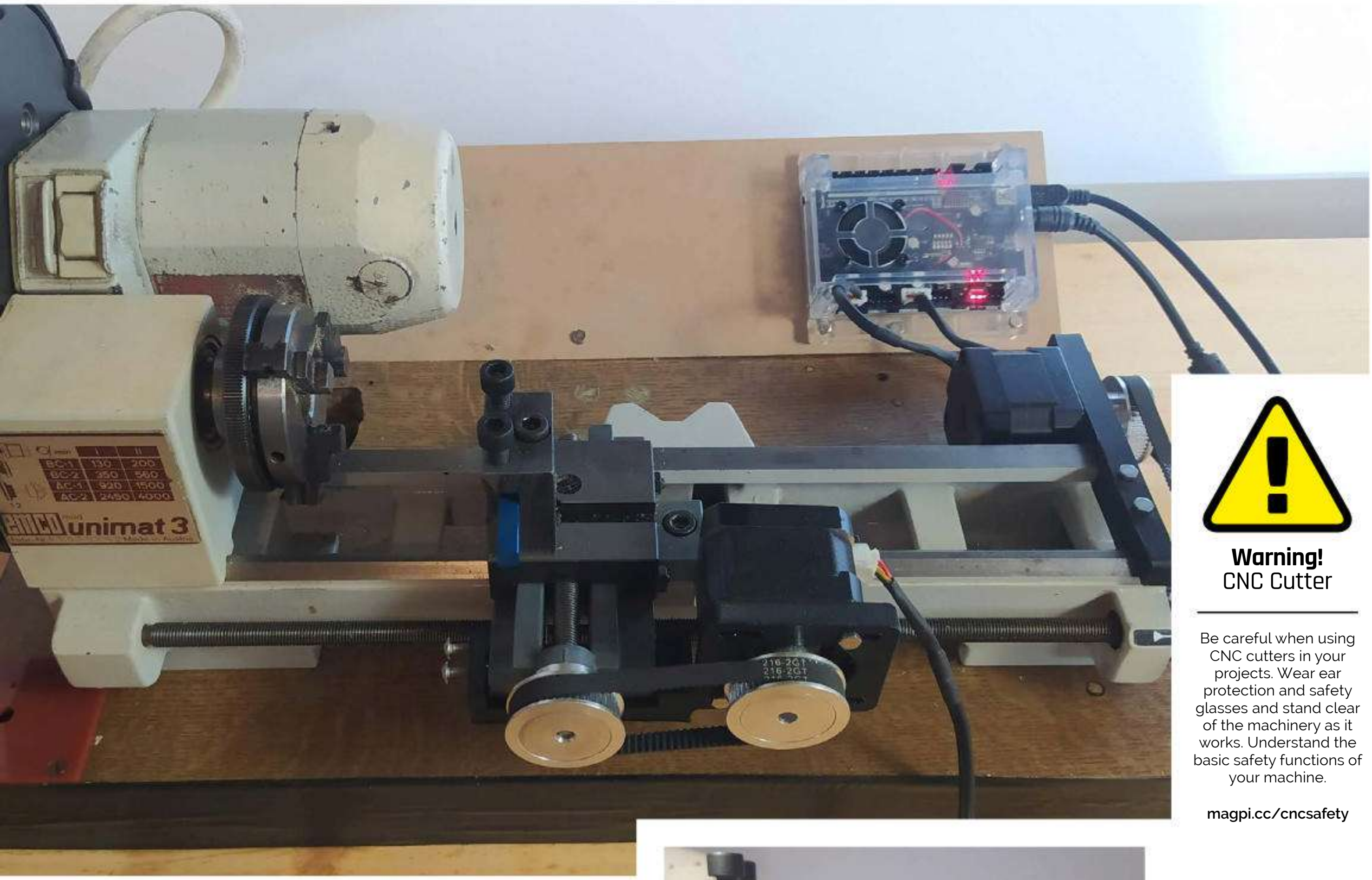


Figure 3
The main ways, or Z axis, complete conversion



Warning!
CNC Cutter

Be careful when using CNC cutters in your projects. Wear ear protection and safety glasses and stand clear of the machinery as it works. Understand the basic safety functions of your machine.

magpi.cc/cncsafety

The two motor mounts clamp onto the various parts of the Unimat using bolts. The 3D printed mount for the main ways (which we will later call the Z axis) was modelled to conform to the profile of the lathe bed and a Nema 17 motor mount was designed at the side. I wasn't 100 percent sure at this stage what size GT2 pulley wheels I would use and in turn what size pulley belt I would use so I designed it with slots so that I could move the Nema 17 position to tension the belt, as in **Figure 3**. The design has a couple of M3 bolt holes and a small block into which I fitted some brass threaded inserts. This clamps inside the end of the lathe bed. I opted for some 48-tooth 30mm diameter GT2 pulley wheels. Removing the Unimat handles from the lead screws, I discovered the shaft where the handles mount had a 5mm diameter, and you can happily buy GT2 pulley wheels with a range of internal bores including 5mm. The motor clamp adaptor for the cross slide simply has bolt holes that allow the adaptor to clamp underneath the cross slide onto the small 5mm height of metal at the edges, see **Figure 4**. While both these solutions seem slightly less than optimal they have proven that they can stay on incredibly well in operation and they aren't really under any forces. →

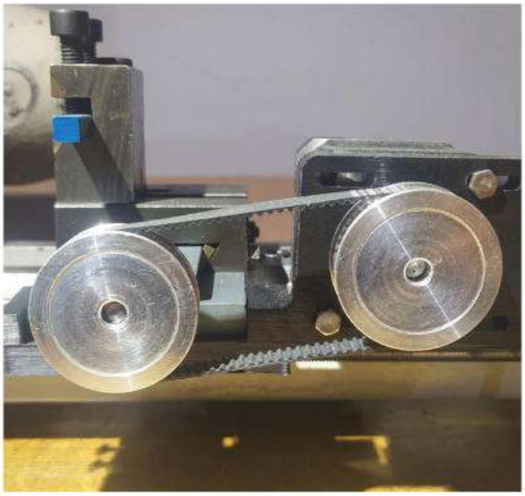


Figure 1 The lathe in question

Figure 4 The cross slide, or X axis complete conversion

> "The design of the printed sections was of course undertaken in FreeCAD"

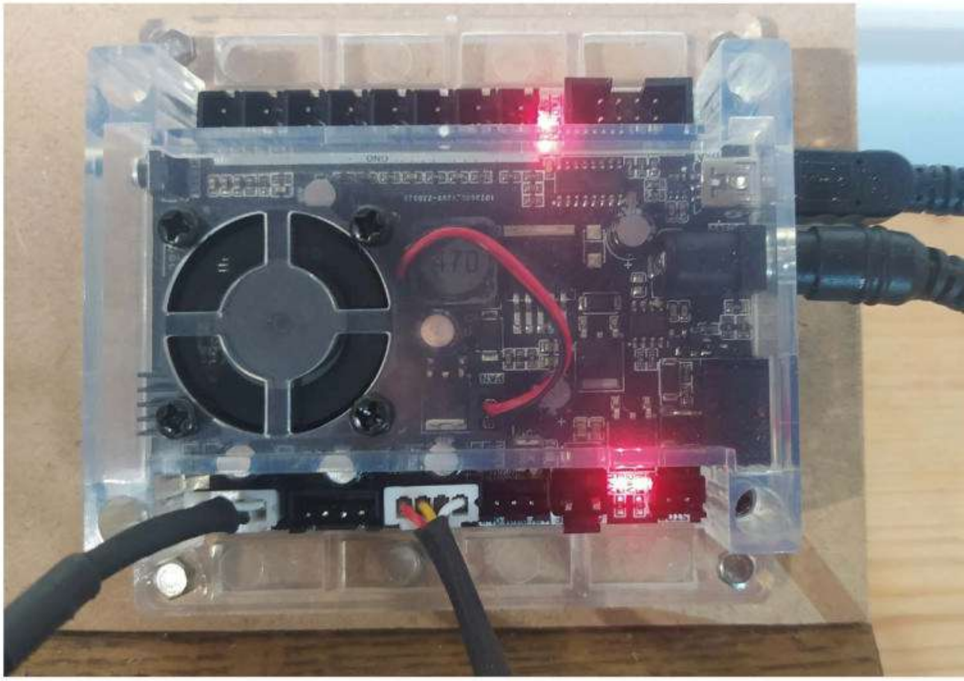
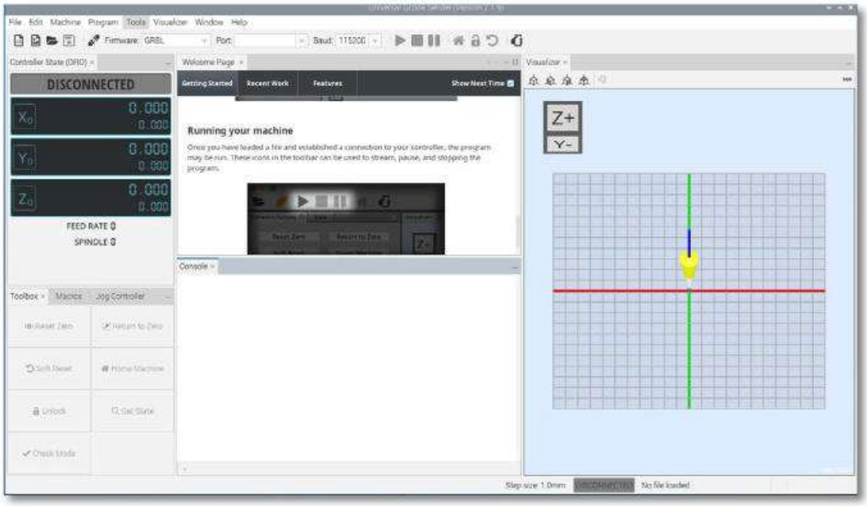


Figure 5
The affordable 'Woodpecker' style boards, often used for the small CNC3018, are a budget way to get up and running with Nema 17 motors

Figure 6
The Universal G Code Sender application running on a Raspberry Pi 400

The design of the printed sections was of course undertaken in FreeCAD and all the files for this project, both the printable stl files and the FreeCAD project files have been posted on Printables. They aren't too complex and are relatively straightforward to print, and they have been printed in PLA with a 60% infill to add some rigidity. The CAD approach was a pretty straightforward sketch then pad/extrude approach using the FreeCAD Part Design and Sketcher workbenches. All the approaches used in making these parts are covered in the free to download FreeCAD for Makers book over on the Raspberry Pi Press site.



Running your machine

Once you have loaded a file and established a connection to your controller, the program may be run. These icons in the toolbar can be used to stream, pause, and stopping the program.

Once printed we assembled the mounts onto the Unimat 3 and then mounted the stepper motors, GT2 pulley wheels and added the short GT2 pulley belts. With the slotted motor mounts allowing for a little tensioning of the belts a 280mm looped belt worked well for the main ways and a 216mm belt worked well for the cross slide adapter.

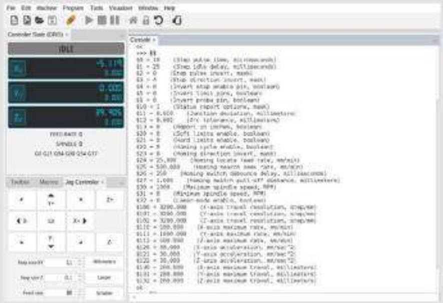
When driving stepper motors there are a lot of options for control systems. One of the simplest to get started with is the open-source GRBL control system, which runs on a variety of microprocessors and can be driven via a choice of G-code sender applications on a variety of platforms. The standard GRBL tends to run on Arduino type platforms but there are a heap of variants and forks of the main project including 32bit versions with support for more than three axes, wireless control and more. I opted to run this project using the very affordable 'woodpecker' control board which is commonly found on budget CNC machines like the CNC3018 style routers. It's worth looking around though; on the list to try is GRBLHal (**hsmag.cc/grblHAL**) which can run on a Raspberry Pi Pico or Pico W.

The next part of the puzzle of a DIY lathe, or indeed any DIY CNC machine, is sending or streaming the G-code to the machine. There are lots of options for GRBL, but one that's nicely supported

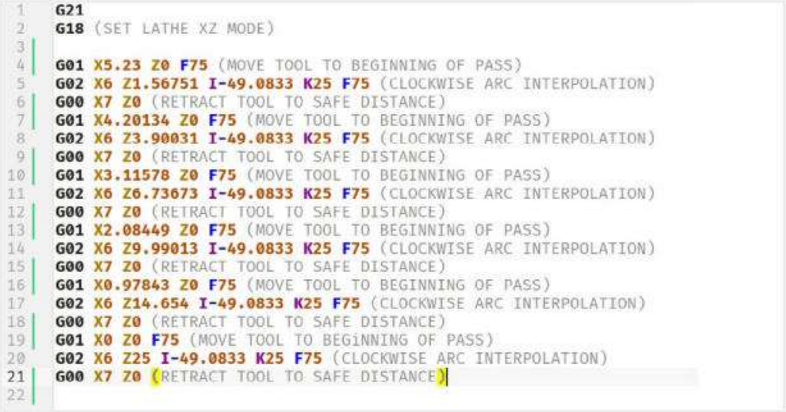
CALIBRATION TIPS

With the stepper motors mounted and connected to the Z and X axis motor drivers on the woodpecker board it was time to calibrate our CNC. This can be a little tricky but you are aiming to work out the number of steps or micro steps the control board needs to send to move the CNC by a known amount. You can set these parameters by connecting the device to a G-code sender, in this case Universal G-code Sender running on a Raspberry Pi 400, and send the "$$" command which shows you the current GRBL settings.

Scrolling through these you should find the $100 $101 and $102 lines which set the steps per mm for the X,Y and Z axis in turn. To alter these you simply type "$100= XXX" replacing XXX with your value. You can calculate these values if you know the thread pitch of your lead screw system, the number of steps per revolution your stepper takes and the microstepping multiplier of your control system. Alternatively you can also work this out empirically by connecting some kind of accurate gauge on the machine (a dial test indicator is very useful, but a set of cheap digital callipers may suffice) set a value and test to see how far the machine actually moves. Through a process of trial and error you should be able to get an accurate repeatable value.



> When driving stepper motors there are a lot of options for control systems



```
G21
G18 (SET LATHE XZ MODE)

G01 X5.23 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z1.56751 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)
G01 X4.20134 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z3.90031 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)
G01 X3.11578 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z6.73673 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)
G01 X2.08449 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z9.99013 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)
G01 X0.97843 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z14.654 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)
G01 X0 Z0 F75 (MOVE TOOL TO BEGINNING OF PASS)
G02 X6 Z25 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL TO SAFE DISTANCE)

```

Figure 7 An example of a G-code file containing multiple arc passes to create the geometry in Figure 8

by Raspberry Pi OS is "UGS", or the Universal G-code Sender.

To set up UGS on the Raspberry Pi head over to the downloads page (**hsmag.cc/G-CodeDownloads**) and download the Linux arm 64 version (presuming you are running the 64-bit Raspberry Pi OS). This will download a .tar.gz compressed archive. Create a folder on your Raspberry Pi, name it **UGS**, and then copy the file you downloaded to it. Next, right-click the tar.gz file you downloaded and extract the archive into the folder. Inside the new extracted folder you will find a folder labelled **bin** containing a file named **ugsplatform**. Right-click, and in the Properties window check this file is set as executable. Double-clicking this file will then launch the UGS platform on your Raspberry Pi, as in **Figure 6**.

With more common machines like 3D printers and CNC routers there are software environments that will create G-code files for our projects. However, it is totally possible to create custom G-code files with little more than a text editor. This approach not only gives you masses of flexibility, but also promotes a deeper understanding of CNC machines making →

custom machine builds and operation more approachable. See **Figure 7**.

There are a lot of G-code references online, but as a small primer let's work through some common G-code commands we used to play with our lathe. First, at the top of a G-code document it's useful to state whether you want to use metric or imperial values for your G-code file. G21 and G20 set G-code to metric (mm) and imperial (in) respectively, so in our header section we write G21.

> "We've set up our lathe so that the Z axis increases in value as the tool moves towards the lathe chuck"

If we were operating a milling machine/router or a 3D printer then it's pretty clear which axis/planes are X Y and Z. With CNC lathes the convention is that the main ways that move the tool towards and away from the chuck towards the lathe tailstock is the Z axis, and the cross slide which passes the tool across the front of the lathe chuck is the X axis. We can set the active plane in the G-code document to X and Z by issuing a G18 command, and this is often referred to as "lathe mode".

Next let's look at getting things moving. G00 is rapid motion, which is for movement through air when the tool is not cutting, if you are familiar with 3D printer terms it's often called "travel". This G-code will move as fast as the machine definitions allow. So if in the GRBL firmware you have your machine set to have 300mm/min as the maximum travel rate then that is how fast the machine will move when a G00 command is sent. You supply it with a set of co-ordinates, in our case just X and Z, but on a three-axis machine it would be X, Y and Z. So for example "G00 X0 Z7" moves the tool to the X axis zero and to the 7mm position in the Z axis.

G01 Linear interpolation. Sometimes referred to as "feed motion", this is the G-code used for motion when the tool is cutting. The feed rate of the machine while running a G01 command is set using the "F" G-code. Most modern machines will interpolate these commands so that if you send a "G01 X10 Y2 X6 F75" then the machine will move in a straight line from its previous position to those co-ordinates at a feed rate of 75mm per minute.

G92 X0 Y0 Z0 is an example of a "Set working offset" command. The G92 command allows us to reset the current co-ordinates of a machine. This means we can set offsets, so for example on a lathe if we place in a piece of round stock that is 12mm diameter and then touch the tool to the outside of the diameter we can then send "G92 X6" so that we have set the current X axis position as 6mm (the radius of the workpiece) away from the 0 origin point. We've set up our lathe so that the Z axis increases in value as the tool moves towards the lathe chuck, so after facing a piece of stock we only have to set the Z axis to 0, sending a "G92 Z0" command. Note that with G92 commands you can send a value to an individual axis, or to multiple axis. If you don't reference an axis within the command it will stay at its previously set value. Note that often we don't include G92 commands inside G-code documents, but rather we can send this command directly to the machine to set a working offset. So we may jog/move a machine into a position and then type into the "command" dialogue/input box in UGS "G92 X12 Z0" and send that command individually.

Figure 8
Using a collection of G02 commands to turn an Ogive shape into a piece of wood

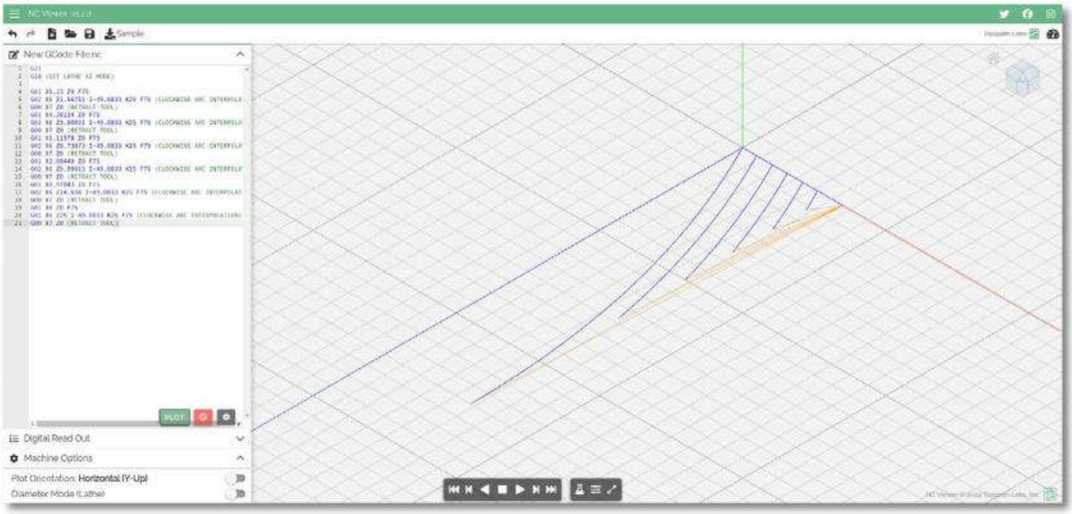
Figure 9
Using an online G-code viewer can help you check your G-codes

One of the advantages of a CNC lathe is the ability to be able to cut arcs and curved paths, as in **Figure 8**. To do this, an important G-code to understand is the G02 (or the G03) commands. G02 can be used to set up a command to follow an arc path clockwise from the tool's previous position and G03 can create an anti-clockwise arc path. In GRBL the G02 and G03 commands begin with the target co-ordinate that the tool will reach at the end of the movement. So to reiterate, the start position is the co-ordinate position of the tool in its last state before the G92 command is called.

So far we have defined the end point of the arc, and we know the origin point for the arc, but we need to add the co-ordinates of the origin point, or more simply the centre point of the circle that defines the arc radius. GRBL uses an additional set of syntax to describe the arc co-ordinates using the letters IJK. In our lathe setup we only are using the X and Z axes, so we only use the I and J to set these co-ordinates. Note that this centre point is defined relative to the starting co-ordinates. Finally, similar to the G01 command, we set the feed rate with an F command.

Armed with these few G-codes it's pretty straightforward to create little programs for your machine. Using a text editor you can type out your commands and save them – G-codes often use the file suffix ".nc" but they are in essence simple text files, so you can write them in a text editor and then swap the file suffix just by renaming the file. A super-handy website that allows you to write and preview G-codes is NC Viewer, seen in **Figure 9**. Testing G-codes there is a safe way to see what happens, and always follow that up with testing the G-code on the machine using 'air cutting', where the machine has no tool or workpiece and is positioned in a way that it should never crash.

The skills learnt in this little project open up all kinds of possibilities for all kinds of machines. Keep an eye out in future issues for projects building custom and more unusual CNC projects!



```
G21
G18 (SET LATHE XZ MODE)

G01 X5.23 Z0 F75
G02 X6 Z1.56751 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLA
G00 X7 Z0 (RETRACT TOOL)
G01 X4.20114 Z0 F75
G02 X6 Z3.90021 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLA
G00 X7 Z0 (RETRACT TOOL)
G01 X3.11578 Z0 F75
G02 X6 Z6.73873 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLA
G00 X7 Z0 (RETRACT TOOL)
G01 X2.00449 Z0 F75
G02 X6 Z9.99913 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLA
G00 X7 Z0 (RETRACT TOOL)
G01 X0.97843 Z0 F75
G02 X6 Z14.604 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLAT
G00 X7 Z0 (RETRACT TOOL)
G01 X0 Z0 F75
G02 X6 Z25 I-49.0833 K25 F75 (CLOCKWISE ARC INTERPOLATION)
G00 X7 Z0 (RETRACT TOOL)
```

Buttons and fastenings

Consider the engineering behind buttons and fastenings – the designs of decades, even centuries, ago still remain true

Nicola King

Nicola King is a freelance writer and sub-editor. 'Buttony' is officially her new favourite hobby, and buttons will now be added to everything she makes!

Above No machine-manufactured buttons here – we're celebrating heritage techniques that need to be passed on. Versatile, customisable, lightweight, artistic... making your own buttons is fun. Change colours, add a festive vibe... it's completely up to you

Try to imagine, for a moment, where we would be without buttons and other textile fastenings. How would we keep our bags secure, our clothing and coats fastened against the elements, or our shoes on our feet?

There are countless types of fastening to hold textiles together, and we cover some of those later in this article. They clearly have an important functional use, but buttons and fastenings don't just connect things. Many also provide a non-utilitarian decorative flair to clothing, for example, enhancing the look of a garment or a bag. Deeper than that, though, is the history, the heritage that we can find by looking at the origins of some of these useful and often beautiful items and, in this tutorial, we are going to delve into the world of button-making – one of the traditional crafts listed on the Heritage Crafts website (**hsmag.cc/heritagecrafts**), a charity that wants to preserve and promote age-old skills. Other 'endangered' crafts include orrery making, cricket bat making, and clock making, to name but a handful.

As a craft, button-making is a very portable and addictive occupation, plus it costs very little to give it a try. No special equipment is required, simply a needle, thread, and a button base of some kind. Once you've made some buttons, there is also plenty you can do with them, the most obvious being to actually use them as buttons! But, any kind of button can also be used for ornamental purposes in home decoration projects, as this imaginative and innovative artist illustrates – **hsmag.cc/augusto**. Handmade buttons can also be made into jewellery items, including earrings, necklaces, and bracelets. Or, how about attaching some larger ones to a string of LED lights for a unique decoration?

> **"Button-making is a very portable and addictive** occupation, plus it costs very little to give it a try"

Living in a button-making hub of yesteryear (Dorset, UK), and a stone's throw from a village that thrived on button-making when the industry was at its height, this author has a very local interest in the topic. So, let's make a button using an age-old technique, hopefully producing something original and unique at the end of the process that also has a function.

THE BYGONE BUTTON INDUSTRY

Dorset buttons were a handmade type of button that originated in Dorset, a county on the south coast of England, and were at their height of popularity between 1622 and 1850. Interestingly, the original Dorset buttons were purely functional – small, white buttons using a very fine thread and thin ring, made for undergarments, and they needed to survive a good wash and scrub, and maybe even a mangle! It was a cottage industry that involved hundreds of families over its around 200-year peak, with skills being passed down through the generations over that period. Workers were remunerated on a piecework basis. It's thought that a key figure in the Dorset button industry was an individual named Abraham Case who set up a button business in Shaftesbury in the early 17th century, making his buttons from things such as fabric, sheep's horn, and thread.

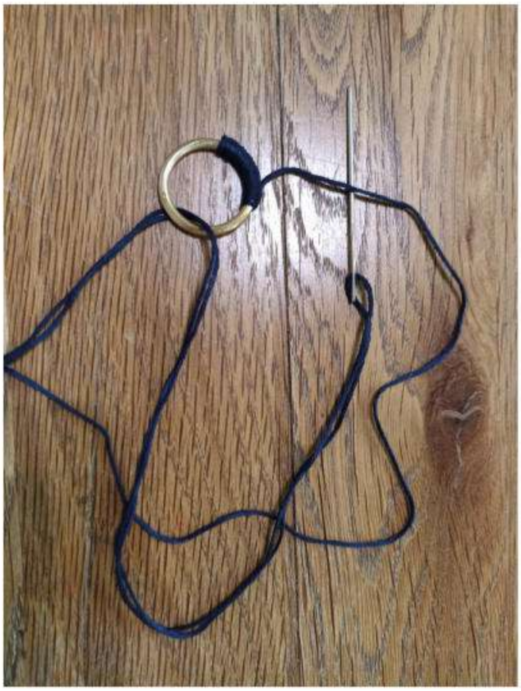
Eventually, machine-made buttons took over, as they did throughout the country once the Industrial Revolution fully took hold, although the decline of button-making in Dorset was very gradual. By the 1850s there was a threat that the skills involved would be lost to machines, but in the early 1900s the Dowager Lady Lees sought to revive the industry in the little village of Lytchett Minster, and began a small but thriving business. Sadly, WWI brought that to an end, but artists and others are keen to keep the craft alive, and many people still make traditional buttons for their own enjoyment. Today, creatives are using Dorset buttons to make hanging ornaments, jewellery, decorations for curtain tie-backs, keychains, and are attaching them to hair accessories, to provide just a few examples.

Dorset buttons were, and still are, made by wrapping a fabric, floss, or yarn around a base material (often made of metal or wood, and now sometimes plastic), and the most famous type of Dorset button is the 'ring' button. There were many traditional designs, and the whimsical names that they were designated included the 'Dorset Knob', the 'High Top', the 'Bird's Eye', and the 'Dorset Cartwheel'.

Dorset Ring Buttons by Gina Barrett is a book that gives an in-depth look at ring buttons, while her *Buttons: A Passementerie Workshop Manual* gives a detailed look at how to make a wide variety of button types. Her website is also worth a long perusal if you want to try your hand at button-making, or 'buttony' as this author prefers, as Gina is a button and passementerie specialist who has also made many authentic buttons for a number of period TV and film projects (**ginabsilkworks.co.uk**).

Of course, while Dorset was one of the most well-known areas for button-making, it was not the only place famous for its button industry in England. Other areas included Birmingham, Macclesfield in Cheshire (which was renowned for its thread-wrapped silk buttons) and Yorkshire.

Below
The blanket stitch is a way of covering the ring before we make the rest of the button. Push the stitches together as you go so that you don't see the ring underneath



BUTTON IT

We suggest using a 25mm sized (or larger) closed brass ring to begin with, as we found it much easier to work with than smaller rings. Bear in mind that the larger the ring, the more thread you will need to wrap it, and the thinner the thread, the more you will need to wrap it also. The first step is to thread a tapestry needle with around two metres of the thread we intend to cover the button with – in this case, an embroidery thread which gives a nice finish and slides easily through the ring. This may seem like a lot of thread to deal with at first, but you do need enough to finish your button in one go if possible, although you can add more thread if you run out. A tapestry needle is used as it has a rounded point that passes easily between thread.

The term used for covering the button base in thread is called 'casting', and involves stitching around the edge of the button with a blanket stitch. Tie one end of the thread to the ring with a single knot, leaving a short tail. Then bring the needle from the back through the middle of the ring, stop, then pass the needle under the thread from right to left (as per **Figure 1**). Pull through and enclose the end of tail as you do so – you'll see small loop forms as you pull it. →

QUICK TIP

If you gently rub the base ring with beeswax before you start a project, it will stop the thread from slipping. A useful tip, especially when you are a beginner.

YOU'LL NEED

- A variety of round-shaped ring bases (eg. a curtain ring)
- A tapestry needle
- Embroidery thread/floss/pearl cotton thread of different colours
- Scissors
- Beeswax (optional)

Below
In Lytchett Minster, Dorset (a short walk from this author's home), you can find this quaint little emporium, Button Shop Antiques, which is appropriately attached to Old Button Cottage. Yes, buttons were big in our neck of the woods!

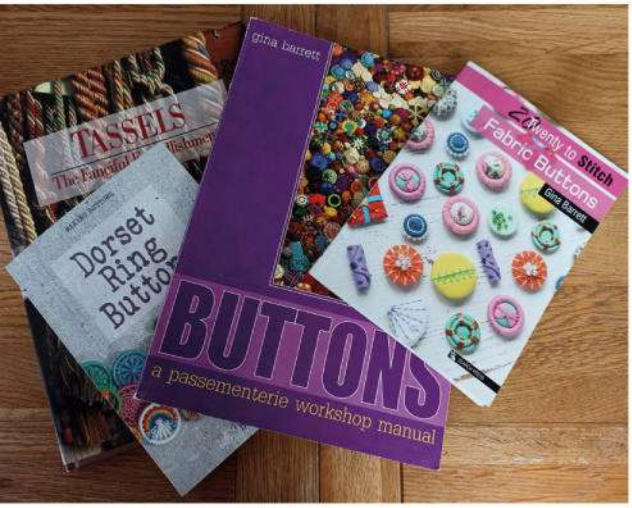
QUICK TIP

If you are using a white or light-coloured yarn or thread, do make sure you have clean hands, otherwise you may discolour the thread and spoil the end product.

Try to pull the thread or yarn quite tightly and you will see a small ridge appearing on the edge as you go. Continue around the ring until it is covered, fully enclosing that short tail end, and do make sure that you keep your stitches close together with no gaps at all, as you should not be able to see the ring underneath. When done, thread the needle through the first stitch to lock it tightly in place, but please do not cut the thread. The next step is something called 'slicking,' which involves moving the edge or ridge that's been created to the back of the ring with your fingers to finesse the look of the button. Simply roll the edge round, which should be very easy to do. Your 'frame' is now ready to make a Dorset button.

SPOKES IN THE WHEEL

We are now going to create the spokes of our button, creating something similar to a type of button called a 'Dorset crosswheel,' and this process is also known as 'laying'. Old buttons of this type had many, many spokes and were usually done in a white thread, but we're going to lay just ten spokes on our button. First, hold your ring so you can see the centre, pull the thread down from the back with your fingers, underneath the bottom of the ring (at the six o'clock position) and then back up to the centre of the top (12 o'clock). You'll need to use your thumb and forefinger to hold the wraps in position. Then slightly rotate the ring to the left (anti-clockwise as we're right-handed, but clockwise if left-handed) and create another spoke in the same way, slightly left of the one already created – we've done this five times which has given us ten spokes. As you do this, you'll notice that the wraps at the back are not in line with the front ones – that's fine, as they now need to be 'centralised,' which essentially just means tidying up the back spokes by bringing your needle from the back through to the front, then down again into the middle of the opposite spoke (creating a stitch if you will). You then need to pull the yarn and the spokes should start to line up. Do this a few times in a few spokes, and use your needle to tweak the back spokes in line with their front counterparts – it's fiddly, but line them up as best you can.



Below You know this author is a fan of a craft book or two… and here are a few more. Search out second hand books online or in charity shops – the older the better! Remember that the techniques behind a number of crafts, particularly heritage crafts, have barely changed over the decades

The next step is to 'fill' the button with a rounding backstitch. To do this, come up from the back to the front at the centre point, then take your needle back over the spokes behind where you came up (that's over both top and bottom spokes which are now lined up). Then take your needle up through between the spokes on the other side of the one you came up initially, so you're moving anti-clockwise, and then take it down between the spokes behind it. You soon get the hang of it – you are backstitching your way round the ring. As you work, keep your middle finger at the back on the thread to keep a good tension. Once you've filled the ring completely, fasten off at the back and cut the thread. As with anything, practice helps, and we found we got much faster with practice. When you get more proficient, you could also add beads as you stitch or change thread colours.

QUICK TIP

Take a look at this helpful YouTube channel for some brilliant button-making guidance: youtube.com/@GinaBSilkworks.

Left
Engineered to solve a problem, each of these fasteners makes our day-to-day lives just that much easier, and they arguably haven't changed much over the decades

QUICK TIP

You can use any type of modern thread or yarn to make these buttons (you don't need to use traditional threads of antiquity) – a thinner thread will have a finer look to it, while thick yarn will make a chunkier button. Silk ribbon or nylon cord give an interesting effect too.

ALL ZIPPED UP

Buttons, and the myriad other fastenings that we use today, tend to be things that we take for granted in our everyday lives, but many of these little wonders have an absorbing and lengthy history that is worth examining. Buttons, in particular, have been around for centuries, and the industry of button-making literally employed thousands of people. We have just scratched the surface here, looking at just one type of Dorset button, but there are many, many designs and a huge number of button types that you can make – from fabric covered to thread wrapped, to lace and embroidered buttons, to name a few. Making buttons is a fun way of connecting with the past, preserving cultural heritage, and celebrating traditional craft skills developed over successive generations.

Left Below
You can also buy inexpensive and useful tools like this and use them to envelop a plastic or metal button in the fabric of your choosing. We simply cut a small circle of fabric slightly larger than the button, pushed the fabric and button front into the depression in the tool, folded the excess fabric into the back of the button front, and then placed the back of the button on top, pressing down with the other part of the tool. Job done!

GET HOOKED

Arguably, one of the handiest types of fastening is the hook-and-loop fastener, sometimes also called a 'touch fastener'.

This type of fastener, or closure system, allows the user to repeatedly fasten and unfasten two surfaces, whether they be the surfaces of a bag, tent flaps, duvet covers, cushions, or even shoes. We're all aware of how it works – this fastener is usually composed of two strips, one featuring tiny hooks, and the other very small loops. When the two are in contact, the hooks catch in the loops and are temporarily but securely attached until pulled apart. But the history of how this fastener came to be is fascinating, and illustrates how nature can inspire the simplest of inventions.

George de Mestral was a Swiss electrical engineer who recognised nature's 'technology' and adapted it, creating the world's first hook-and-loop fastener. The story goes that, back in 1941, George was returning from a walk with his dog when he noticed how the seeds, or burrs, from the burdock plants that he had passed on his walk were attached to himself and his dog because, as he saw under his microscope, the seeds had thousands of tiny hooks which had latched onto the fabric of his clothes. He guessed that if he could recreate this synthetically, he'd have created a useful way of fastening things quickly and easily.

Over a ten-year period, de Mestral perfected his ideas and submitted a patent in Switzerland in 1951 which he eventually succeeded in obtaining in 1955. Velcro IP Holdings LLC, in business as Velcro Companies, was founded by de Mestral in the 1950s, and was the original manufacturer of hook-and-loop fasteners. It's very important to remember that 'VELCRO® Brand' is a brand and not a noun or a verb. Today, many imitations of the original hook-and-loop fastener now exist in the marketplace.

When NASA started using the original hook-and-loop fastener in zero gravity, skiers then started using it in their ski-wear, and interest then snowballed (!) – the hook-and-loop fastener gained in popularity, and today it's used widely in many areas of day-to-day life.

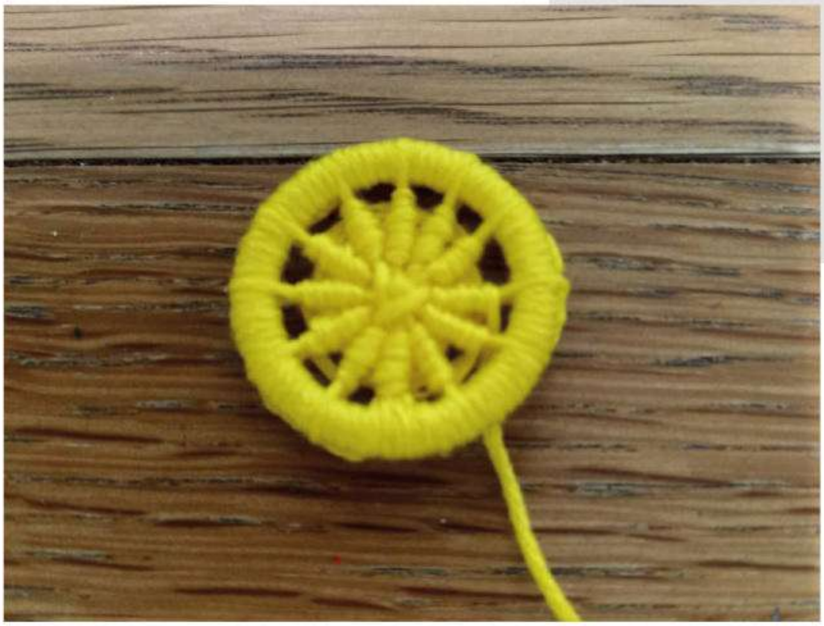
OTHER FABRIC FASTENERS

While buttons are one of the most obvious fastening devices, there are of course a plenitude of other fastenings to consider. Here are just a few:

- Cufflinks: hundreds of year ago, shirt cuffs were held together with threads, strings, and ribbons. It wasn't until the 17th century that the earliest forms of cufflinks emerged. Still popular, wearing cufflinks is thought to show an attention to detail in an outfit.
- Zips: Gideon Sundback, a Swedish/American engineer, is generally credited as the inventor of the zip, or 'zipper'. He had the patent for a 'separable fastener' in 1917. A slider on two rows of metal/plastic teeth which interlock, it is arguably one of the most useful fasteners. Invisible, open-ended, magnetic – there are many variations.
- Toggles: many of us have had a duffel coat with a toggle fastening, which is much like a button, but instead of pushing the button through a hole, we just manoeuvre it through a loop. Instant rustic style and easy to do, even in the freezing cold!
- Safety pins: the modern safety pin was invented back in 1849, while a precursor was arguably the 'fibula' invented around the 14th century BCE for the tunics of ancient Greeks. With a plethora of uses, they are particularly good at securing nappies.
- Snap fasteners or poppers: Heribert Bauer was a German inventor who patented, in 1885, the modern snap fastener. However, there were other inventors around at the time who were also obtaining similar patents. Two interlocking discs fit together until pulled apart, and this invention revolutionised sewing as the snap fastener offers a discreet, hidden closure and gives a clean look to the front of garments.
- Shoe laces: the old adage 'if it ain't broke, don't fix it' applies to many types of fastening that have an extremely old heritage. Shoelaces are one of those fasteners that arguably haven't actually changed that much over the centuries. Evidence of shoelaces made from animal hide has been found dating back to around 3000 BCE, and medieval footwear had laces that passed through hooks and eyelets. Arguably, the only difference between those more primitive laces and today's versions is in the innovative materials now used to create them. A simple fastener, but one that helps your footwear give proper support and stability.

> **As you work, keep your middle finger at the back** on the thread to keep a good tension

Below
Once again, this author was lucky enough to find a very inexpensive second hand book or two on the subject, but visit your local library or charity shop to see what you can find



DigiKey
DigiKey

ONLY THE BEST

LED Lighting

Light up your projects, and your world, with an array of LEDs

By **Phil King**

Let there be light! Whether you want to create mood lighting for areas of your home or add some blinkenlights to projects, LEDs are the key ingredient. They come in many different types and form factors, from long flexible strips to matrix grids – also good for digital signage – and a range of shapes. In this Best of Breed, we're throwing a little illumination on LED lighting options.

While flashing a solitary LED is the Hello World of physical computing, there are far more interesting things you can do once you have a bunch of LEDs to play with. Single-colour LEDs have their uses, but for the best results you'll want some RGB (or RGBW) ones that can light up in any shade. Ideally, each LED should be individually addressable for maximum control and the ability to create impressive animated effects such as sparkles and colour sweeps.

Of course, you'll need a way to control them, be that with a Raspberry Pi computer, Pico, or other microcontroller. Whatever you use, programming your own patterns and effects is a lot of fun.

Cosmic Unicorn vs RGB LED Matrix Panel

PIMORONI £80 / $89 | pimoroni.com

ADAFRUIT £22 / $30 | adafruit.com

With a Raspberry Pi Pico W on board, this all-in-one LED matrix panel works out of the box, with no need to add separate hardware to drive it. Just connect it up to a computer via USB and start programming using the C/C++ or MicroPython libraries. Coming with several code examples, the latter enables you to easily display shapes, sprites, and a selection of fonts.

Arranged in a square 32×32 grid with 6mm pitch, the 1024 addressable RGB LED 3.5mm square pixels have built in diffusion. Working alongside Pico, 12 FM6047 constant current LED drivers control them at around 300fps at 14-bit resolution, so there's no hint of flicker.

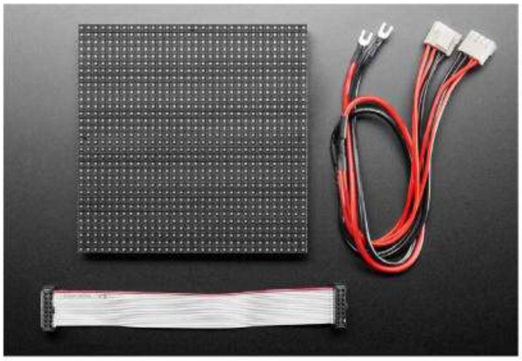
Pico W's Wi-Fi and Bluetooth connectivity also open up extra possibilities for controlling the panel remotely from another device. Other bonus features include a mini 1W audio speaker – so you could turn it into a Bluetooth speaker with audio visualiser – and a couple of Qwiic/STEMMA ports to connect breakouts.

Left This all-in-one 32×32 RGB LED matrix has a Pico W soldered onto the rear

Below This HUB75 panel is 'dumb', but comes with cables

While considerably less expensive than the Cosmic Unicorn, this is a 'dumb' panel with no memory nor self-drawing capability, so you'll need extra hardware to drive it, such as Adafruit's RGB Matrix Bonnet for Raspberry Pi, or a suitable microcontroller such as the RP2040-based Interstate 75. The LED matrix panel is available in various resolutions, up to 64×64, and pitch measurements between the pixels – between 2.5mm and 6mm.

These HUB75 standard panels – many others are available from various suppliers – are the sort that can be chained together to make a larger 'video wall', provided you have enough processing power and RAM. Adafruit's guide (**hsmag.cc/ledmatrixguide**) explains the basics of how they work. Note that



there's no PWM control to vary the LEDs' brightness, so you need to continually redraw the whole matrix at high speeds, requiring a fast controller (50Hz or more) for the best results. Typically, you'll also need to supply 5V power separately to the panel.

VERDICT

Cosmic Unicorn

An all-in-one Pico-powered panel that's very versatile.

9/10

VERDICT

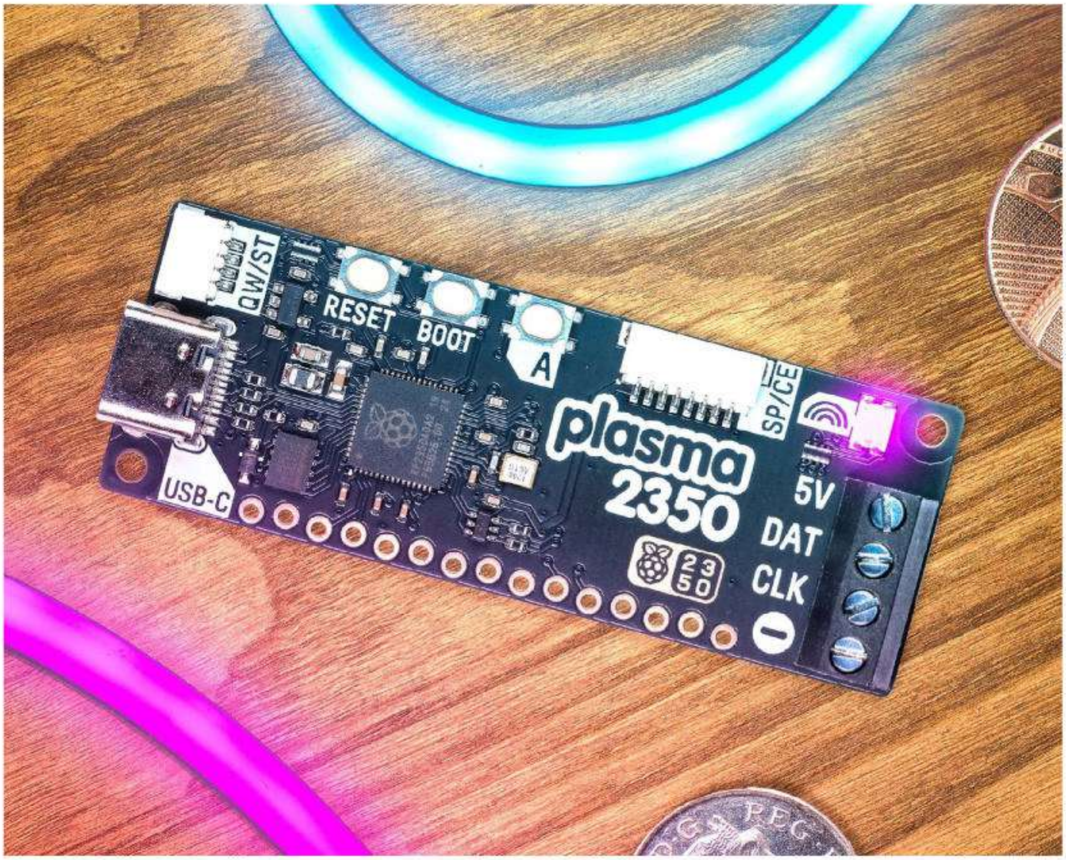
RGB LED Matrix Panel

Good value and chainable, but requires extra hardware to drive it.

8/10

Plasma 2350

PIMORONI £12/$13 | pimoroni.com



Left This tiny LED controller board packs in quite a few features

If you want to light up a string or strip of WS2812/NeoPixel or APA102/DotStar RGB LEDS, this tiny board makes it incredibly easy. Also available as a starter kit with a USB-A to USB-C cable and 10m string of 66 individually addressable frosted LED stars, it can deliver enough current to power much longer strings.

Programming it in MicroPython is made simpler by a custom UF2 file and several code examples – you can also use most of those for the Plasma 2040 and Plasma Stick, as they use the same library. Impressive effects include falling snowflakes, alternating/random blinkies, sparkles, fire, pulsing, and a sweeping rainbow.

Based around the same RP2350A microcontroller chip as Raspberry Pi Pico 2, the Plasma 2350 has plenty of processing power to handle complex patterns with no flicker. While there's no wireless connectivity, bonus features include a GPIO breakout header, Qwiic/STEMMA QT and SP/CE connectors, plus user buttons.

VERDICT

Plasma 2350

A great solution An easy way to control NeoPixel/ DotStar LED strings.

9 /10

Neon-like RGB LED Strip with Diffuser vs NeoPixel Ring – 24 × 5050 RGB LED

PIMORONI ◈ £18/$20 | pimoroni.com

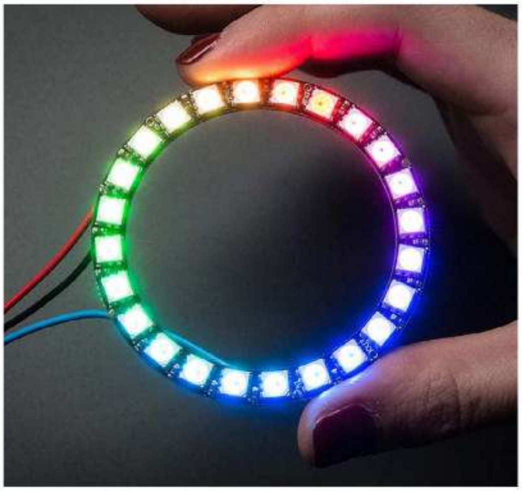
ADAFRUIT ◈ £13/$17 | adafruit.com

There are many options when it comes to buying long strips – which usually come on a spool – of addressable RGB LEDs, which are ideal for festive lights and general mood lighting. Things to watch out for include the pixel density (LEDs per metre), brightness, power requirements, and water/ingress resistance if using them outdoors.

This one from Pimoroni is sold in 1m strips of 96 LEDs. The big selling feature is that it's encapsulated in chunky silicone which – as well as offering IP65 standard water resistance – diffuses light to produce the illusion of glowing neon with no obviously visible gaps. In addition, the narrow sides are translucent, allowing more light through. Like most strips, it's chainable and there are extra power/ground wires if you need to inject some extra 'juice' – each metre requires 15W of 5V power

Left ◈ In real life, the LED pixels aren't as noticeable as they are in this photo

Below ◈ With ultra-bright RGB LEDS, this is one dazzling ring



Ideas for using an LED ring include as a bike headlight or in a lamp. This particular model has 24 '5050' RGB LEDs – the ultra-bright kind often used in smart strip lighting – arranged in a 2.6-inch (outer) diameter circle. Adafruit sells a whole range of other sizes with varying numbers of LEDs, all with integrated drivers and chainable. For a larger ring, you can even buy quarter-circles of 15 RGB or RGBW LEDs to make a 6.2-inch diameter circle of 60.

Whichever ring you choose, you can program lighting effects in Python or CircuitPython with the aid of Adafruit's CircuitPython NeoPixel module. In CircuitPython, you can also make use of Adafruit's LED Animations library, which includes effects such as comet, theatre chase, pulse, blink, colour cycle, rainbow, and sparkle.

VERDICT

Neon-like RGB LED Strip with Diffuser

Perfect for when you want that glowing neon effect.

9/10

VERDICT

NeoPixel Ring – 24 × 5050 RGB LED

Easily controllable, chainable, and very bright.

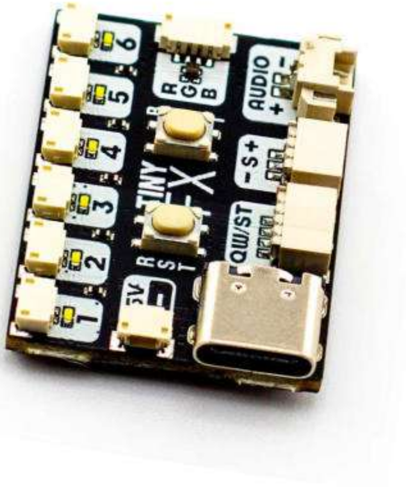
8/10

Tiny FX

PIMORONI £15/$17 | pimoroni.com

As Sparkfun rightfully declares: "Sometimes we want our projects on, but sometimes we want to turn them off for a while to save power". And this is where the Nano Power Timer comes into play. Consuming a minimal amount of power, approximately 35nA, you can easily add a simple timer to your project.

The board features a TPL5110 delay IC which is configured based on the input resistance. Sparkfun has added a series of switches to the board, and depending on how you turn them on and off is how you set the delay. This board can handle voltages from 1.8 to 5.4 and can set an on/off delay of between 100ms to two hours. If you're designing a circuit that needs intermittent power, this is definitely a board you should check out.



Left
Sometimes good things come in small packages

Below
The multicoloured matrix can produce some dazzling displays

VERDICT

Tiny FX

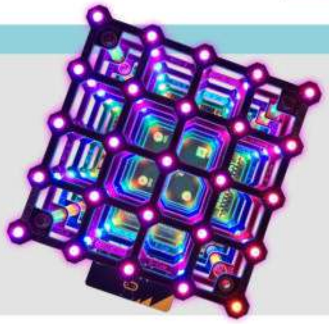
A tiny board with a surprisingly large feature set.

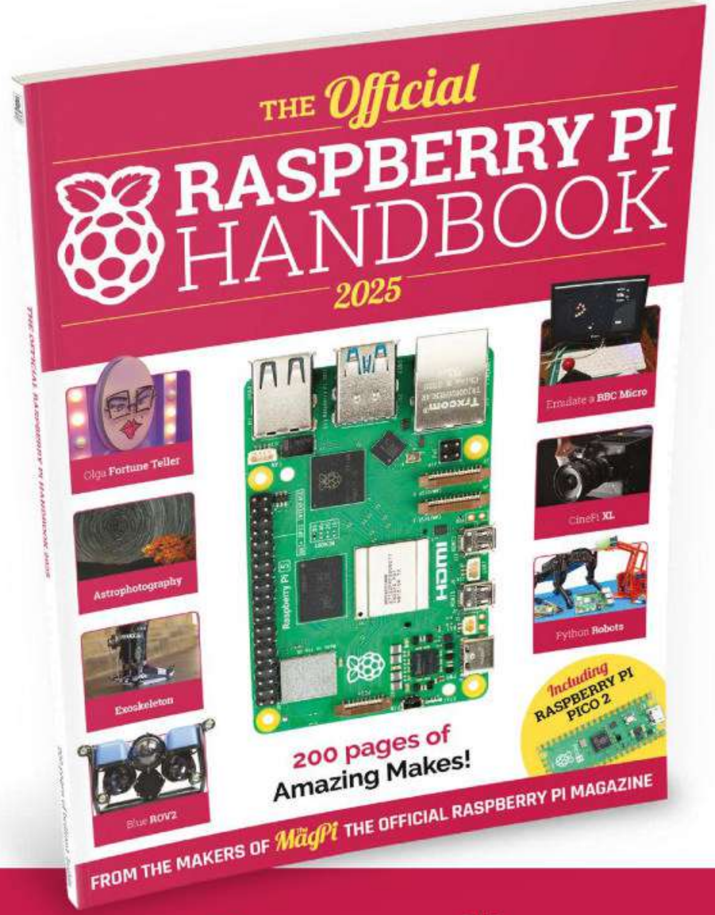
8 /10

CUBE:BIT

4TRONIX £23/$31 | 4tronix.co.uk

For something a little different, how about this RGB NeoPixel cube? It comes in 3×3×3, 4×4×4, and 5×5×5 versions, the last two coming with the Base section required to power it and connect a Raspberry Pi Zero (or micro:bit) – see **hsmag.cc/cubebitguide** for instructions and Python example code. Easy assembly involves connecting the 'slices' with metal stand-offs.




THE Official
RASPBERRY PI HANDBOOK
2025
Olga Fortune Teller
Astrophotography
Exoskeleton
Blue ROV2
Emulate a BBC Micro
CineFi XL
Python Robots
Including RASPBERRY PI PICO 2
200 pages of Amazing Makes!
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

RASPBERRY PI ADVENTURE
A TOUR OF RASPBERRY PI 5

Olga The Fortune Teller

HANDHELD GAMING

with Raspberry Pi

Retro gaming on the move can be fun and creative. PJ Evans grabs some spare batteries

Maker

PJ Evans

PJ is a writer, software engineer and a dab-hand at Galaxians. When not using the original machinery he can be found 'doing research' for this feature.

mastodon.social/@mrpjevans

Whether you're looking for a nostalgia hit or just an authentic way of playing the classics, handheld gaming with Raspberry Pi is fun, fun, fun. Not only can you play retro games galore but also get involved in the creation of 8-bit-style games. With the increasing power of Raspberry Pi Zero and Raspberry Pi Pico (especially with the new Pico 2), there's never been a better time to put down your phone and grab something a bit more tactile. Over the next few pages we'll look at your options for handhelds, setting them up and how to write your own games.

Warning!
Copyright!

All the software discussed in this feature has been made freely available by its respective copyright holders. Please be cautious when finding ROMs for your retro setup that you are not downloading from a site hosting software illegally.

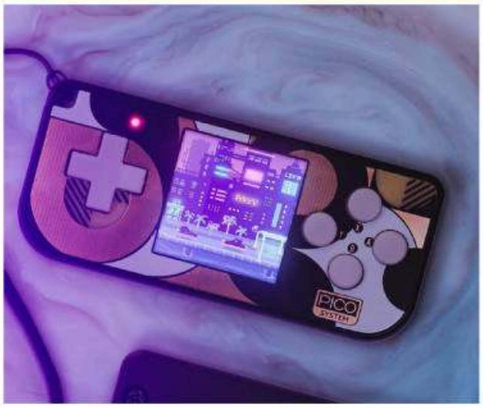
magpi.cc/legalroms

RETROFLAG GPI CASE 2W

£75
magpi.cc/gpicase2w

With more than a passing resemblance to a classic 1990s device, the GPi Case is a stunning little handheld. This version is powered by Raspberry Pi Zero 2W, so it can handle more demanding emulators than earlier versions. A 3in IPS screen and audio jack provide the full experience.

>> **Perfect for**
The dedicated retro gamer



PIMORONI PICOSYSTEM

£59
magpi.cc/picosystem

Want to go smaller? The cute little PicoSystem is a Pico-based handheld targeting those wanting to try their hand at writing their own games. It's small enough to feature a lanyard hole, and built from solid aluminium so will handle bumps and bruises. Pimoroni provides all the info to get started and loads of sample games.

>> **Perfect for**
Hacking about and writing your own games

NULL 2

£37 / $49
null2.co.uk

Fancy something a little more hackable? The Null2 is a self-assembly kit in the classic landscape form factor. It's ideal for those who want to customise and endlessly tinker. Powered by a Raspberry Pi Zero 2 it comes with all the parts you need for a tiny handheld experience.

>> **Perfect for**
Those who love to game and hack in equal measure

KITRONIK ZIP96

£36
magpi.cc/zip96

Here's something a bit different. The ZIP96 is a perfect introduction to coding through game creation. A super-simple LED array powered by Raspberry Pi Pico complete with control buttons. Kitronik has thoughtfully created a series of lesson plans to show how to write simple games like *Snake* for this low-cost device.

>> **Perfect for**
The younger games coder

PI GRRL 2

Varies
magpi.cc/pigrrl2

How about building your own handheld from scratch? If you have access to a 3D printer and know which end of a soldering iron gets hot then you can have a lot of fun putting this kit together. The result is a cute Raspberry Pi Zero-powered handheld ready to run your favourite games.

>> **Perfect for**
Retro-crazy makers

RETRO LITE CM4

Varies
magpi.cc/retrolitecm4

If you really want to make a statement, and have some proper building skills, try the Retro Lite project. A beautiful larger handheld in a format similar to the Nintendo Switch and Steam Deck. It requires custom PCBs and a lot of work to build, but the Compute Module 4 powered result will be mighty.

>> **Perfect for**
The serious retro gamer and maker

> “If you have access to a 3D printer and know which end of a soldering iron gets hot then you can have a lot of fun”

WHAT'S THE SOFT OPTION?

You've got your hardware but what's the best OS to use?

Recalbox

recalbox.com

Over one hundred systems are supported out of the box and installation is a breeze. Ideal if you just want to get on and start gaming.

RetroPie

retropie.org.uk

For a long time, this was the de-facto standard for Raspberry Pi gaming. It's not updated as often now, but still worth a look.

Batocera

batocera.org

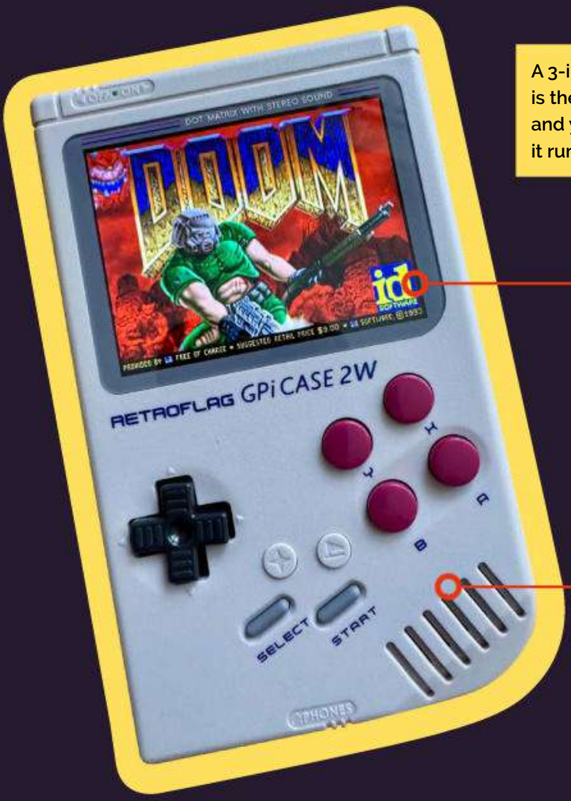
New kid on the block, Batocera supports over 190 systems and is gaining a reputation as an excellent platform.

Build the ultimate retro gaming handheld

Play thousands of games on the move with Retroflag's amazing GPi Case and a Raspberry Pi Zero 2W

You'll Need

- Raspberry Pi Zero 2 (No headers) **magpi.cc/zero2w**
- Retroflag GPi Case 2W **magpi.cc/gpicase2w**

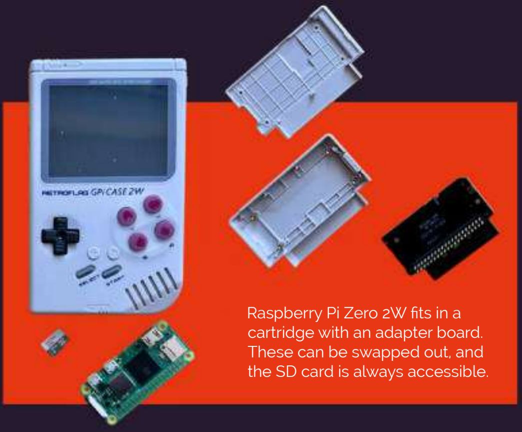


A 3-inch IPS screen is the perfect size and yes, of course it runs *Doom*.

The GPi Case recreates that 1990s handheld gaming feel, with a few extra buttons

▼ Recalbox comes with loads of free (and legal) games for you to play, including this ZX Spectrum scrolling shooter.

Getting started in retro gaming has gotten much easier over the years. More powerful hardware means emulators run better too. Retroflag have been at the forefront of the retro hardware scene with a series of handheld and console cases for Raspberry Pi, some of which bear tribute to some of the legendary devices of the 1980s and '90s. Combine the latest handheld case with the power of a Raspberry Pi Zero 2W and you have something really special. Getting started can be intimidating so here's a short guide to creating the best handheld retro gaming device out there.



Raspberry Pi Zero 2W fits in a cartridge with an adapter board. These can be swapped out, and the SD card is always accessible.

01 Get Recalbox

Recalbox is the leading platform for retro gaming on Raspberry Pi (and other systems too). The developers have excelled in making a true plug-and-play system that removes a lot of the endless config editing and moving strange files around to make things work. In fact, there are over 100 systems supported right out of the box. Better yet, they have created a build especially for the GPi case range. Start by getting a microSD card and writing the Recalbox OS image to it. You'll find it in Raspberry Pi Imager > Emulation and game OS > Recalbox > GPICase2W. *Do not* hook up the Raspberry Pi to a monitor and boot.

02 Assemble the cartridge

The Raspberry Pi Zero 2W fits into a cartridge that can be removed from the main body (you can buy additional carts too). The instructions will ask you to install additional software, but the Recalbox image already has everything already installed. If you boot to a regular HDMI monitor, it will mess things up for later. Instead, remove the microSD card and follow the assembly instructions to attach the Raspberry Pi Zero 2W to the adaptor and then mount both in the cart without the card. Screw the cart halves together so the adapter's pogo pins are married to the GPIO.

03 First boot

Before placing the cart into the back, flip the 'Safe Shutdown' switch to 'On' and make sure the internal battery is fully charged or keep the USB power cable attached. Place the cart firmly in the slot and switch on. The first boot will run through a configuration process but there are no questions to answer. Just let it do its thing (it'll give you some great tips too) and you'll be taken to the systems menu. A number of platforms are already available with games included, so you can start right away. Use the D-pad to navigate, B to select and A to go back. When done with a game, press START and the flag button above it together.

04 Connect to Wi-Fi

Now you have a base system up and running, you'll probably want to add more games. The sensible option is to use Wi-Fi. On the main system menu, press the SELECT button to access the main menu. Now scroll down to 'Network Settings' and press B. Scroll to 'Enable Wi-fi' and press B. Now you can go to 'Network' and choose your local Wi-Fi network. When selected go to 'Wi-fi Password' and enter your password using the groovy keyboard system. You can also use WPS if your router supports it. Finally, choose a hostname for your new toy and you should be able to find it on your local network.

05 Upload ROMs

If you want to add additional games to your handheld, Recalbox has made it easier than just about any other retro gaming platform. When we connected to Wi-Fi, the device started sharing a directory on the network. All you need to do is connect to that folder just like any other network share. On Windows you can use the network browser or 'Map network drive' option. On macOS, Finder will see the handheld automatically. Raspberry Pi and Linux users can mount the filesystem like any other. To add games for a particular system, navigate to 'roms' and then find the system name you want and drop the files there.

06 Refresh and scrape

You won't be able to see your new games right away. The system needs to rescan and restart first. From the Systems menu, go to the system you want to update. Now press START and scroll down to 'Update games list'. Once the update is complete, you can now automatically add artwork and metadata by using the scraper. From the Systems menu, press START and scroll to 'Scraper'. Starting the scraper will scan all your games and add artwork automatically. It can take some time but you can target individual systems in the menu first. Now, it's time to get playing.

»Top Tip«

Safe shutdown

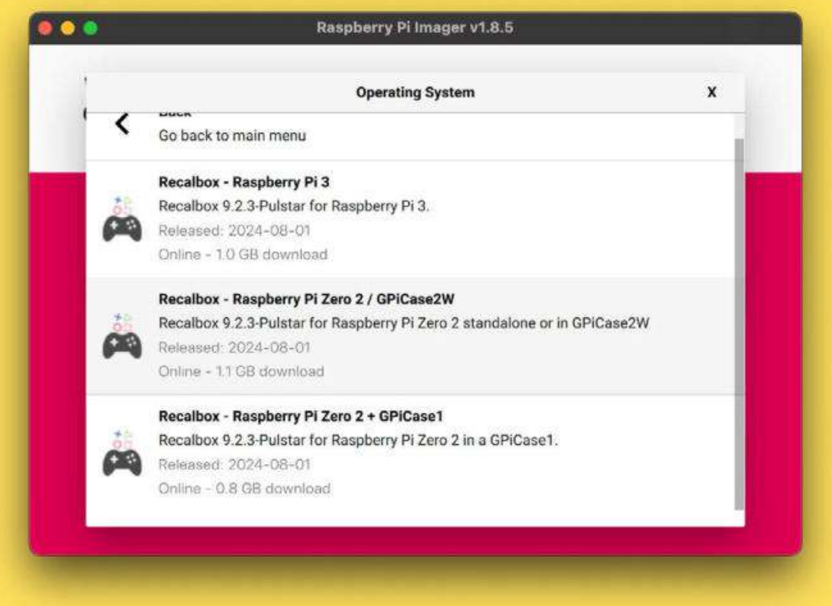
Recalbox supports the GPi Case's safe shutdown mode. When enabled, using the case's power switch will perform a controlled shutdown, protecting your precious handheld.

»Top Tip«

Other GPi cases

If you can't find an image for the combination of Raspberry Pi and GPi case you are using, check Recalbox's own site: **recalbox.com**

▼ Raspberry Pi Imager contains all you need to install Recalbox onto a microSD card



Recalbox - Raspberry Pi 3
Recalbox 9.2.3-Pulstar for Raspberry Pi 3.
Released: 2024-08-01
Online - 1.0 GB download

Recalbox - Raspberry Pi Zero 2 / GPiCase2W
Recalbox 9.2.3-Pulstar for Raspberry Pi Zero 2 standalone or in GPiCase2W
Released: 2024-08-01
Online - 1.1 GB download

Recalbox - Raspberry Pi Zero 2 + GPiCase1
Recalbox 9.2.3-Pulstar for Raspberry Pi Zero 2 in a GPiCase1.
Released: 2024-08-01
Online - 0.8 GB download

Make your own handheld games

with PicoSystem

What's better than playing games? Making your own!

You'll Need

- Pimoroni PicoSystem **magpi.cc/picosystem**
- Laptop or desktop computer
- USB-C cable

Live like it's the '90s with your own version of *Snake* on a 120 x 120 display

Pimoroni's MicroPython libraries make reading control input as simple as possible

Pimoroni's PicoSystem is a unique and fun little beastie. Comparable only with Panic's PlayDate, PicoSystem is a small handheld gaming platform that's aimed at those who want to write games as much as play them. It's powered by the mini but mighty RP2040 and features a full-colour 1.6in display, D-pad and four buttons. You can write games in super-fast C++ or more friendly MicroPython. Getting started can be daunting, but the excellent documentation and software tools take away a lot of the pain. Here we'll create a simple *Snake*-like game in MicroPython.

> “Aimed at those who want to write games as much as play them”

01 Set up the PicoSystem

Like any Raspberry Pi Pico, you can change the firmware on the PicoSystem by putting it in bootloader mode and updating the firmware. The first thing we need to do is upload the Micropython firmware, as PicoSystem ships with a C++ stack. You can download the latest version from **magpi.cc/picosysrelease**. You need to download '**pimoroni-picosystem-{version}-micropython.uf2**' (where 'version' is a group of numbers). Connect the PicoSystem to your computer, hold down X and switch it on. A directory will be mounted just like a USB drive. Copy the uf2 file into the directory. It will then automatically unmount. You're ready to go.

02 Create your code

Let's write *Snake*! PicoSystem works on an infinite loop, each iteration of the loop being known as a 'tick'. Every time the loop runs, we move the snake in the chosen direction, check the direction hasn't changed and also check if the snake has collided with food, the wall or itself. To enter the code, we recommend using Thonny (**thonny.org**) which will recognise the PicoSystem when connected and allow you to add files. Create a new file on the PicoSystem called **main.py** and enter the code as shown on p106. You can also download the code from **magpi.cc/picosyssnake**. Save the file onto the PicoSystem and then disconnect.

03 Make it your own

Let's play *Snake*! If a file called **main.py** is present on the PicoSystem, it will automatically run on boot. Switch the PicoSystem on and your game should start immediately. Grow your snake by eating the food (green block) and see how long you can make the snake without hitting the edge of the screen or itself. Now you have a basic game running, how can you make it better? Could speed slowly increase? Multiple food to collect? How about a high-score system or better graphics? *Snake* is a great starting point for your own game-writing adventures.

»Top Tip«

Read up!

PicoSystem comes with great documentation for both its C++ and MicroPython platforms: **magpi.cc/picosysdocs**

▼ **PicoSystem game:** *Rocks and Diamonds*

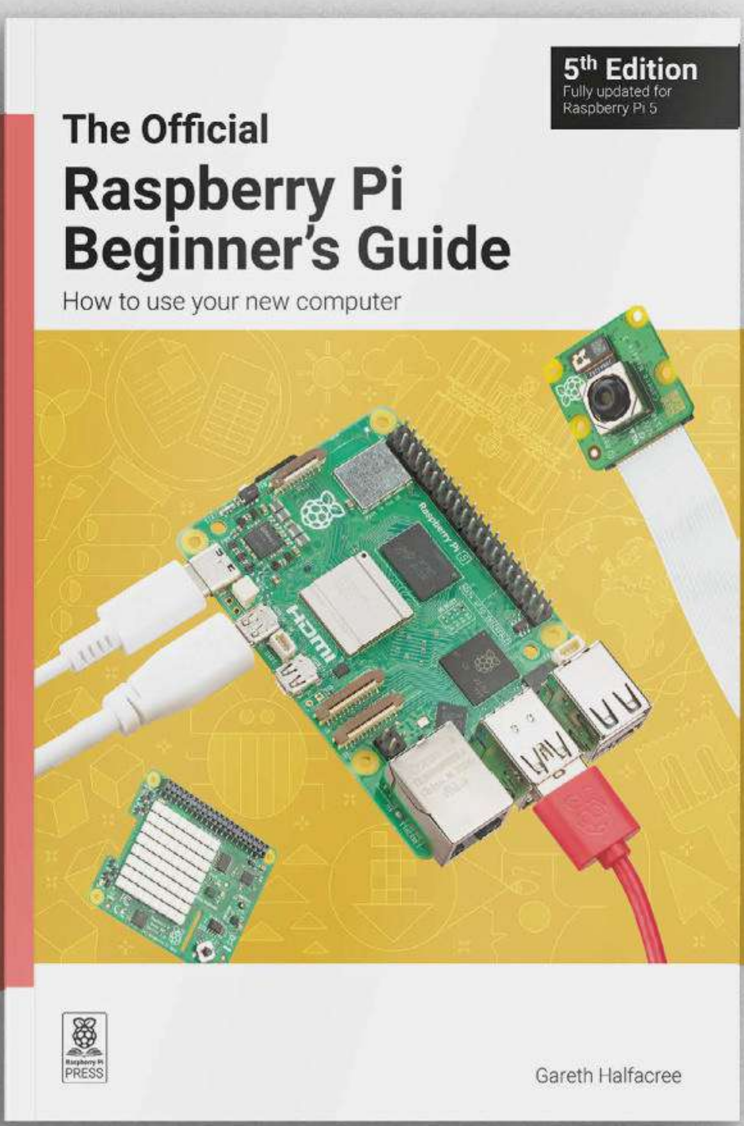
▲ **PicoSystem game:** *Super Square Bros*

main.py

DOWNLOAD THE FULL CODE: magpi.cc/picosyssnake

> Language: **MicroPython**

```python
import utime
import urandom

# Reset everything
pen();cursor();camera();clip();alpha()

# Reset so we have a 2-block snake and food
def game_reset():
    global snake, move
    snake = [[55,60], [60,60]]
    move = [5, 0]
    add_food()

# Add new food in a random place
def add_food():
    global food_x, food_y
    food_x = urandom.randint(0, 23) * 5
    food_y = urandom.randint(0, 23) * 5

def update(tick):
    global direction, move, snake

    # Detect a change in direction
    if pressed(UP): move = [0, -5]
    elif pressed(RIGHT): move = [5, 0]
    elif pressed(DOWN): move = [0, 5]
    elif pressed(LEFT): move = [-5, -0]
    head = snake[len(snake) - 1]
    new_head = [head[0] + move[0], head[1] + move[1]]

    # Remove the tail
    snake.pop(0)

    # Check whether the new ‘head’ will collide with the rest of the body
    for segment in snake:
        if new_head[0] is segment[0] and new_head[1] is segment[1]:
            game_reset()
            return

    # Has the snake reached edge of the screen?
    if new_head[0] is 0 or new_head[0] is 120 or new_head[1] is 0 or new_head[1] is 120:
        game_reset()

    # Has the snake eaten some food?
    if new_head[0] is food_x and new_head[1] is food_y:
        snake.append([head[0] + move[0], head[1] + move[1]])
        add_food()

    # Add the head
    snake.append(new_head)

def draw(tick):
    # Clear the screen
    pen(0, 0, 0)
    clear()
    pen(15, 15, 15)
    flip()

    # Draw each segment of the snake
    for segment in snake:
        frect(segment[0], segment[1], 5, 5)

    # Draw the food
    pen(0, 15, 0)
    frect(food_x, food_y, 5, 5)

    # Controls the speed of the game
    utime.sleep(0.2)

# Start the game
game_reset()
start()
```


5th Edition
Fully updated for Raspberry Pi 5
The Official
Raspberry Pi
Beginner's Guide
How to use your new computer
Raspberry Pi PRESS
Gareth Halfacree

Pibo

▶ Circul US ▶ magpi.cc/pibo ▶ £851 / $1,114

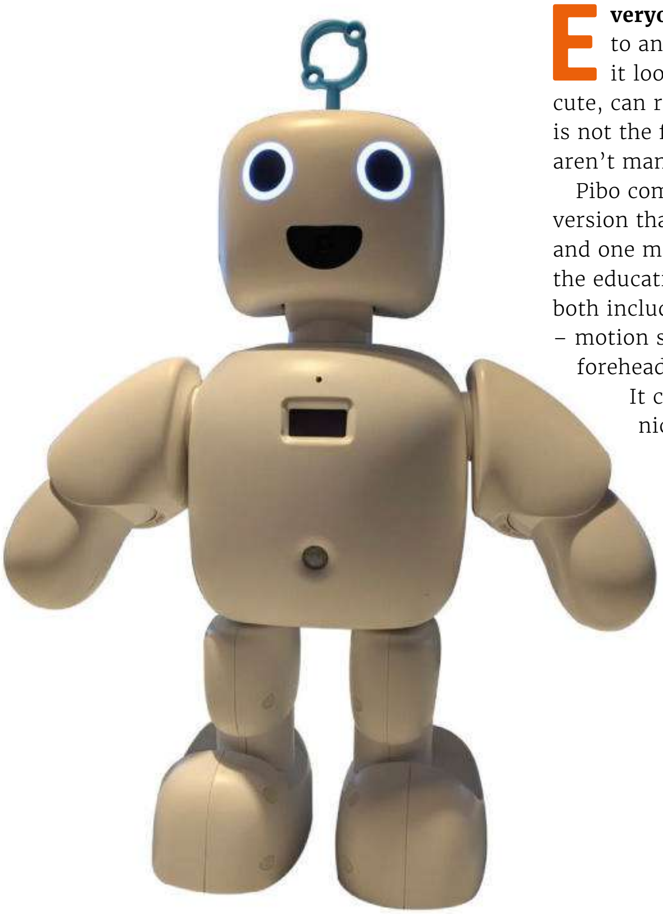
This cute and smart robot can be an AI assistant or an educational coding tool. **Rob Zwetsloot** connects with it

SPECS

DIMENSIONS: 250(w) × 395(h) × 125(d) mm, 2.2kg

INPUTS: Touch sensor, MEMS microphone, PIR sensor, USB 2.0 port

OUTPUTS: 2x speakers, 128x64 OLED display, USB 2.0 port



▶ A cute little robot – well, it's 40cm tall which isn't that little

Everyone loves a bipedal robot. Being able to anthropomorphise an automaton so that it looks more friendly, and in this case very cute, can really help with interaction. While Pibo is not the first bipedal Raspberry Pi robot, there aren't many of them and none as advanced as it.

Pibo comes in two versions – an AI assistant version that you can talk with and interact, and one more suited to education. We received the educational version for review, however both include the same features and hardware – motion sensors, camera, touch sensitive forehead, etc.

It comes fully assembled, which is very nice as putting together the various motors and other components together correctly has been a pain with similar products in the past. All you need to do is turn it on and get it connected to your Wi-Fi network, either via a wireless access point the robot creates, or via a wired connection if you have a USB to Ethernet adapter handy.

The whole thing is powered by a Raspberry Pi Compute Module 4, so it has plenty of oomph – especially needed for the computer vision and voice recognition tasks.

Verdict

A cute and very easy to use robot with a ton of functionality that will take some time to fully discover.

9/10

I have control

The robot itself is made in Korea, and most of the surrounding documentation and such are in Korean as a result. However, the tools and IDE (integrated development environment) can be switched to English just fine, and we didn't experience any language issues.

The tools allow you to play around with the various functions of the robot. Changing the colours of the eyes (independently if you wish), checking if the motion-sensing and touch inputs are working, recording sounds, playing sounds, moving the various motors – you can get a great feel for what the robot can do. With a solid grasp of this, you can then start programming the robot in the IDE.

There's a couple of programming methods – one is a block-based flow a little like NODE-Red, which also helps you understand the coding logic and variables of Pibo, and then there's the Python programming mode which allows for full control.

The functionality is huge, and we were really impressed by the object detection built into the camera. We also like making little messages and images on small LED screens, so having interactive elements that worked with the 128x64 display scratched a specific itch for us.

Learning for all ages

While the whole system may not be useful to teach people on their very first steps into coding, or even maybe robotics, it's a great next step thanks to its intuitive design that lets you play with its features, and block based programming that can lead into Python. The price is a little hefty, and some English features are still incoming, but we had a great time using Pibo either way – one for the little desk display we think.

> “The functionality is huge, and we were really impressed by the object detection built into the camera”

▲ Pibo comes pre-made in this fancy briefcase. Just pop on the antenna

HatDrive! Nano

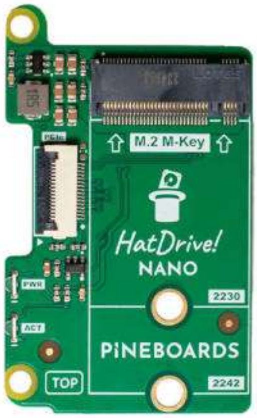
▶ Pineboards ▶ **magpi.cc/hatdrivenano** ▶ £9 / $10

Smaller than your average M.2 HAT for Raspberry Pi 5. By **Phil King**

SPECS

FEATURES: Power and activity lights; 25mm PCIe FPC cable; 3A buck converter

SSD SIZES: M.2 2230, M.2 2242



▲ The diminutive HatDrive! Nano works with M.2 2230 and 2242 size SSDs

▶ It will fit into the official Raspberry Pi 5 case along with an optional Active Cooler

Big doesn't always mean better, as proven by the HatDrive! Nano, which at 55×34mm is about half the size of most M.2 HATs. This means it'll fit inside the official Raspberry Pi 5 case, albeit with the fan section removed.

The good news is that, elevated by a couple of standoffs, the Nano can work alongside an Active Cooler. Some other M.2 HATs can do likewise, but this smaller board leaves most of the Cooler's fan unobstructed so cooling performance should be slightly improved. The GPIO pins and camera/display ports are also easy to access.

A tiny ribbon cable is used to connect the Nano to Raspberry Pi 5's PCIe slot, but you'll need to make sure it's the right way around (with the arrow symbol next to the Nano's PWR light) otherwise it won't work.

Drive on

With everything connected and a suitable SSD slotted in, it's time to switch on. Then it's the usual procedure of formatting the SSD with Imager, and installing an OS if you want to boot from it – in which case you'll need to change the boot order in raspi-config's Advanced options. You'll also want to enable the faster PCI Gen 3 mode by adding a line (`dtparam=pciex1_gen=3`) to the bottom of the **/boot/firmware/config.txt** file.

> Elevated by a couple of standoffs, the Nano can work alongside an Active Cooler

Performance is similar to other M.2 HATs, depending on the SSD used. You can boot up from SSD in around 10 seconds, while we measured the read and write speeds (with PCIe Gen 3 enabled) at around 860MB/s and 670MB/s respectively.

Verdict

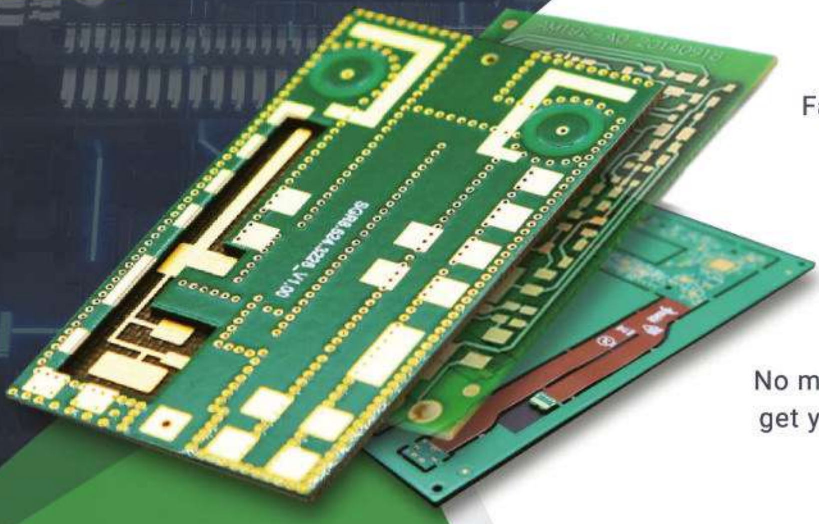
An inexpensive, smaller M.2 HAT with no loss in performance.

8/10

PCBWay

EMPOWERING AI INNOVATION, ONE PCB AT A TIME

In the rapidly advancing world of AI, precision and reliability are essential. PCBWay provides high-quality PCB prototyping, assembly, CNC machining, and 3D printing services to bring your AI-driven projects to life. From robotics to smart devices, we offer a complete solution to power your next AI breakthrough.



Precision PCBs

Perfect for the complex designs that drive AI hardware.

Rapid Prototyping

Fast turnaround times as 24 hours to keep your innovations on schedule.

Global Reach

No matter where your lab or workshop is, we'll get your boards to you quickly and affordably.

End-to-End Support

We're more than manufacturers; we sponsor and collaborate with AI developers, startups, and makerspaces worldwide.

Why Choose Us?

With PCBWay as your partner, you can focus on pushing the boundaries of AI, while we ensure your hardware is up to the challenge.

Start building the future of AI today!

Visit PCBWay.com to discover how our PCBs can power your next AI innovation.

service@pcbway.com

www.pcbway.com

10 amazing: Raspberry Pi 5 cases

Keep your Raspberry Pi 5 safe with these excellent enclosures

It's nice to dress up, even if you're a microcomputer like Raspberry Pi 5. There are many options to choose from too – from traditional cases that keep Raspberry Pi 5 suitable for your desk or under your media centre, to retro console look-alikes and all-in-one screen/tablet cases. Here are ten we like, some of which you can make with a 3D printer.

▲ Official case

Eben approved

A simple yet stylish and effective case, this layered plastic enclosure also includes an active cooling fan and also comes in black.

magpi.cc/case | £10 / $10

▲ 3D-printed case

3D simplicity

This 3D-printed case even comes with a button you can use with the actual Raspberry Pi 5 power button, space for the Active Cooler, and plenty of ventilation.

magpi.cc/rpi53dcase | Free

▶ Swap Pi 3D-printed case

Imitation is flattery

Based on the gorgeous Fractal Designs North Pi, this is a 3D-printed version of the case that normally uses wood, with some mechanical differences.

magpi.cc/swappi | Free

▼ Argon ONE V3 with M.2 NVME

Brick house

This beefy metal case includes a special M.2 NVME connection that differs from the M.2 HAT for Raspberry Pi. It also grants access to the GPIO pins.

magpi.cc/argononev3 | £46 / $49

▲ Modular 3D-printed case

Customisable at home

This cool case doesn't need supports and is constructed with screws to make for a very aesthetically pleasing design, complete with Active Cooler support.

magpi.cc/modular3d I Free

▼ Argon NEO 5

Metallic simplicity

A pared-down case compared to the Argon ONE, the NEO 5 is no less hardy, with its own case fan to keep Raspberry Pi cool too.

magpi.cc/neo5 I £18 / $19

◀ SmartiPi Touch Pro W

Display case

This case allows you to easily attach a screen to Raspberry Pi 5, and the stand means you can put it just about anywhere. It does not come with a screen included though.

magpi.cc/smartiprow I £27 / $36

▼ Pibow Coupe 5

Sandwich case

The Pimoroni classic has been refreshed for Raspberry Pi, with plenty of room to add the Active Cooler to your Raspberry Pi 5 too.

magpi.cc/pibow I £10 / $13

▼ Pironman 5

Mini PC

We love the emergence of classic PC towers used to house a Raspberry Pi – complete with gamer lights for some extra authenticity.

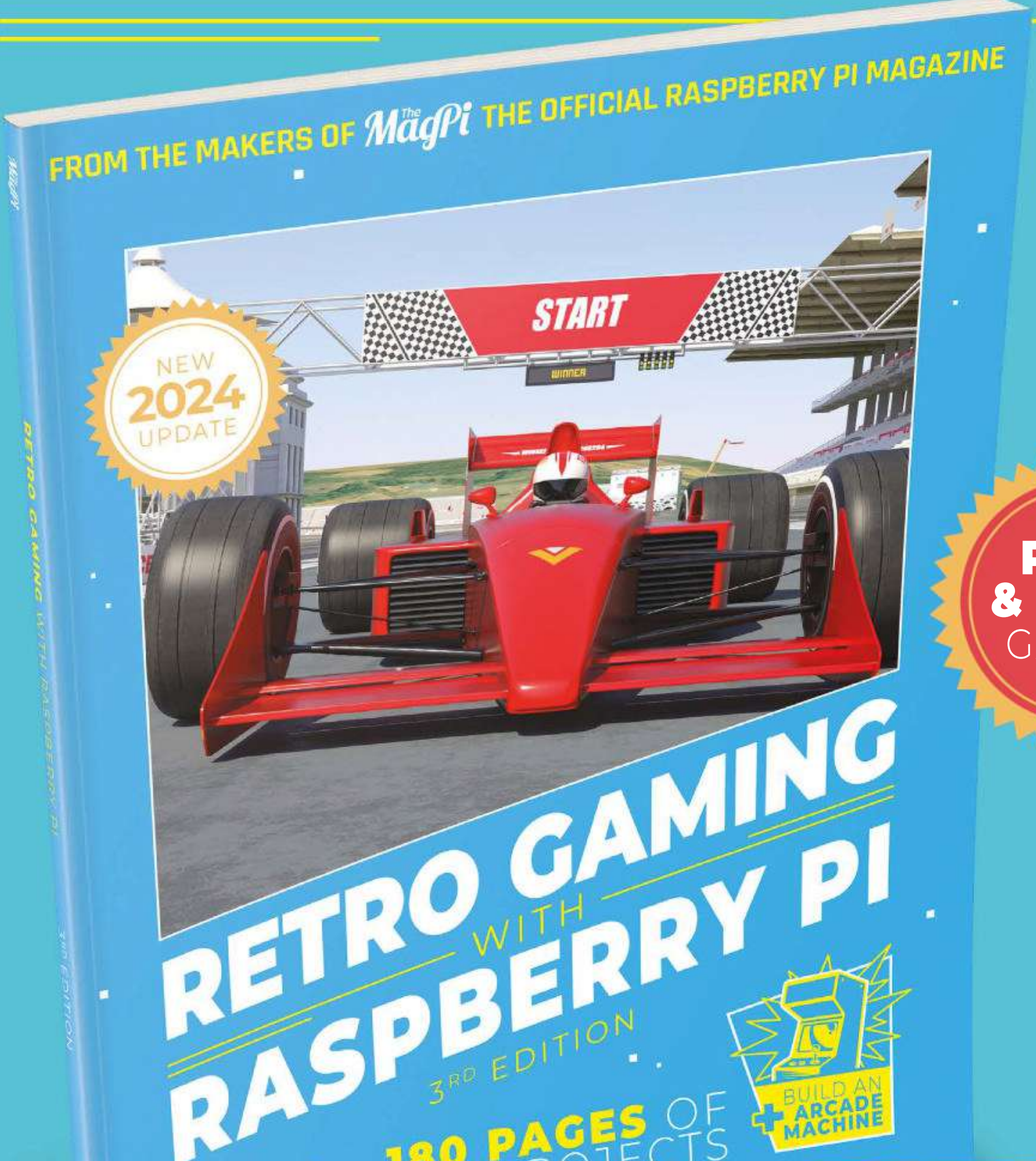
magpi.cc/pironman5 I £60 / $80

▼ Game 5Pi Retro Gaming Case

Nostalgic throwback

This case comes in a square, like the consoles of old, which gives it extra room for ventilation. The flap comes up too, revealing the USB and Ethernet ports.

magpi.cc/game5pi I £17 / $20


FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
NEW 2024 UPDATE
START
WINNER
RETRO GAMING WITH RASPBERRY PI
3RD EDITION
180 PAGES OF VIDEO GAME PROJECTS
BUILD AN ARCADE MACHINE
RETRO GAMING WITH RASPBERRY PI 3RD EDITION

PLAY & CODE GAMES!

Display the list of the files found in the selected VMU.

Kari Lawler

A YouTuber with a passion for collecting and fixing classic computers, as well as retro gaming

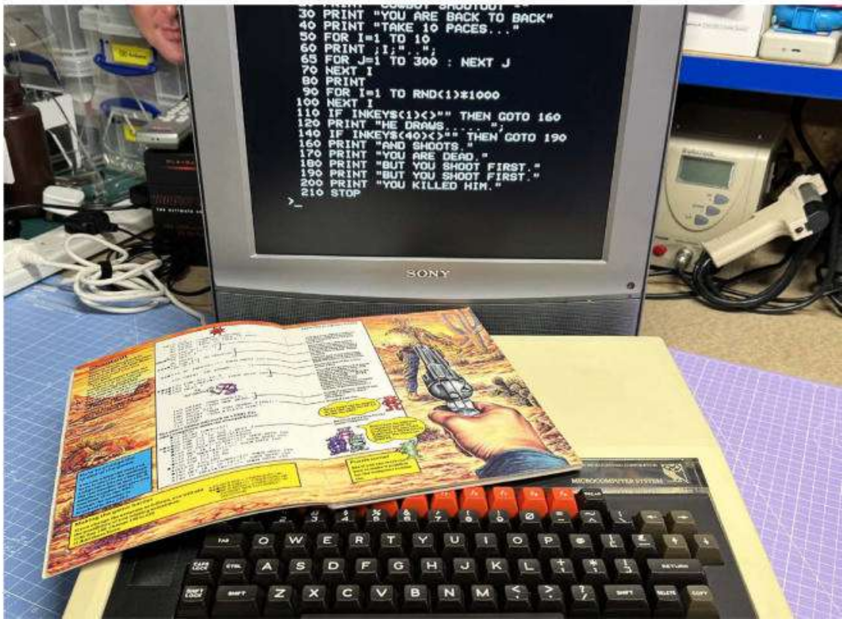
> Name **Kari Lawler** | > Occupation **Tech entrepeneur and consultant**
> Community role **YouTuber** | > URL **magpi.cc/kari**

Kari Lawer has a passion for retro tech – and despite being 21, her idea of retro fits with just about everyone's definition, as she collects and restores old Commodore 64s, Amiga A500s, and Atari 2600s. Stuff from before even Features Ed Rob was born, and he's rapidly approaching 40. Kari has been involved in the tech scene for ten years though, doing much more than make videos on '80s computers.

"I got my break into tech at around 11 years old, when I hacked together my very own virtual assistant and gained some publicity," Kari says. "This inspired me to learn more, especially everything I could about artificial intelligence. Through this I created my very own youth programme, called Youth4AI, in which I engaged with and taught thousands of young people about AI. As well as my youth programme, I was lucky enough to work on many AI projects and branch out into government advisory work as well. Culminating, at 18 years old, in being entered into the West Midlands Women in Tech Hall of Fame, with a Lifetime Achievement Award of all things."

What's your history with making?

Being brought up in a family of makers, I suppose it was inevitable I got the bug as well. From an early age, I fondly remember being surrounded by arts and crafts, and attending many sessions. From sewing to pottery and basic electronics to soldering, I enjoyed everything



```
30 PRINT "YOU ARE BACK TO BACK"
40 PRINT "TAKE 10 PACES..."
50 FOR I=1 TO 10
60 PRINT ;I;"..";
65 FOR J=1 TO 300 : NEXT J
70 NEXT I
80 PRINT
90 FOR I=1 TO RND(1)*1000
100 NEXT I
110 IF INKEY$(1)<>"" THEN GOTO 160
120 PRINT "HE DRAWS.....";
140 IF INKEY$(40)<>"" THEN GOTO 190
160 PRINT "AND SHOOTS."
170 PRINT "YOU ARE DEAD."
180 PRINT "BUT YOU SHOOT FIRST."
190 PRINT "BUT YOU SHOOT FIRST."
200 PRINT "YOU KILLED HIM."
210 STOP
>_
```

▶ Fun fact: the code listings in *The MagPi* are inspired by magazines from the 1980s which also printed code listings. Although you can download all of ours as well

◀ One of Kari's more recent projects was fixing a PSOne, the smaller release of the original PlayStation but with a screen attached

▼ Recreating classic NES Tetrinomoes with a 3D printer to make cool geometric magnets

I did. Which resulted in me creating many projects, from a working flux capacitor (well, it lit up) for school homework, to utilising what I learned to make fun projects to share with others when I volunteered at my local Raspberry Pi Jam. Additionally, at around the age of 12 I was introduced to the wonderful world of 3D printing and I've utilised that knowledge in many of the projects I've shared online. Starting with the well received 24 makes for Christmas I did over on X [formerly Twitter] in 2017, aged 14, which featured everything from coding Minecraft to robot sprouts. And I've been sharing what I make over on my socials ever since.

How did you get into retro gaming?

Both my uncle and dad had a computer store in the '90s, the PS1/N64 era, and while they have never displayed any of it, what was left of the shop was packed up and put into storage. And, me being me, I was quite interested in learning more about what was in those boxes. Additionally, I grew up with a working BBC Micro in the house, so have fond memories playing various games on it, especially *Hangman* – I think I was really into spelling bees at that point. So, with that and the abundance of being surrounded by old tech, I really got into learning about the history of computing and gaming. Which led me to getting the collecting bug, and to start adding to the collection myself so I could experience more and more tech from the past.

What's your favourite video that you've made?

Now that's a hard one to answer. But if I go back to one of my first videos, Coding Games Like It's The '80s, it's one that resonates with how I got my first interest in programming. My dad introduced me to Usborne computer books from the 1980s, just after I started learning Python, and said 'try and convert some of these'. I accepted that challenge, and that's what got me fascinated with '80s programming books, hence the video I made. With the Usborne books specifically, there is artwork and a back story for each game. And while technically not great games, I just love how they explain the code and challenge the reader to improve. For which, I'm sure some of my viewers will be pleased to hear, I have in the works more videos exploring programming books/magazine type-in listings from the '80s.

> At around the age of 12 I was introduced to the wonderful world of 3D printing

MagPi Monday

Amazing projects direct from social media!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they have made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!

01. The train system gets better and better! We love this final (?) product

02. Oh, we're excited to see what this ends up being?

03. Stephen has several Pico 2 boards to use – together or separately?

04. For some reason we could not escape these in the 2000s. Apparently they're back, and Pico-powered

05. PiMowBot testing on 'indoor lawns'. Don't trim the carpet too low

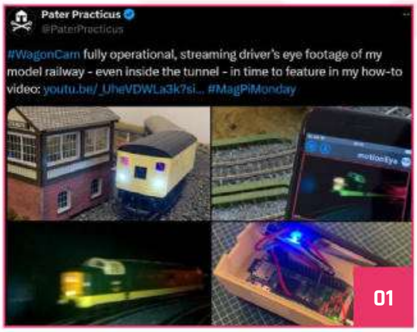
06. Kev is a braver person than us to publicly pit Python and C++ against each other, but running Assembler on Pico 2 is great

07. A modular Raspberry Pi Pico kit that can also be a robot sounds fantastic to us

08. A neat little project with CircuitPythonr

09. Really appreciate that this camera mount attaches directly to Raspberry Pi – very smart

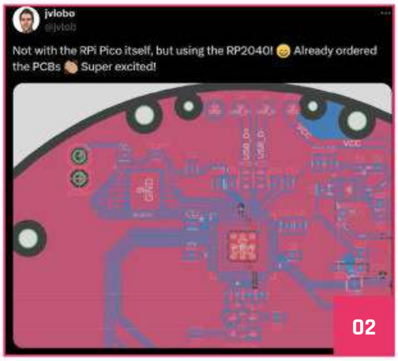
10. We do appreciate that there are a lot of different GUI libraries for Python to play around with



Pater Practicus
@PaterPracticus

#WagonCam fully operational, streaming driver's eye footage of my model railway - even inside the tunnel - in time to feature in my how-to video: youtu.be/_UheVDWLa3k?si... #MagPiMonday

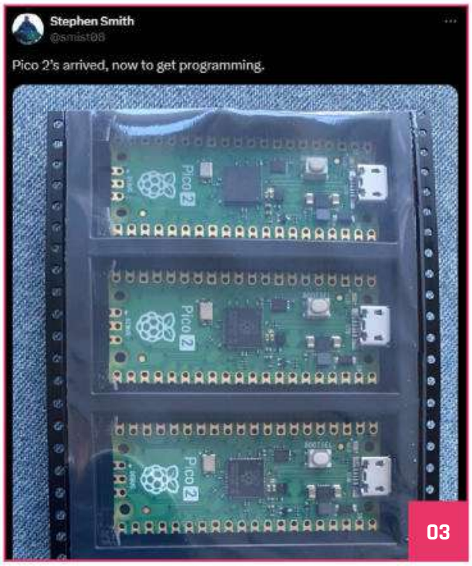
01



jvlobo
@jvlob

Not with the RPi Pico itself, but using the RP2040! 😊 Already ordered the PCBs 👏 Super excited!

02



Stephen Smith
@smist08

Pico 2's arrived, now to get programming.

03

04

Bailey Townsend
@FatFingers23

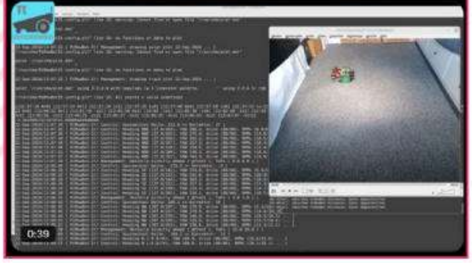
I picked up an old Robosapien toy from a flea market and got it to listen to commands via a Raspberry Pico.

05

TGD
@TGD_Consulting

Happy #MagPiMonday to all robotics fans and a big thank you @ullisun58 for sharing his test run of the latest release of the #PiMowBot software with his MINI on his indoor lawn test field.

#make #RPi #smart #3dprinted #IoT #lawn mower #robot



06

kevinmcaleer 23/09/2024

Happy MagPiMonday - this weekend I hosted a Smackdown between C+ Micropython

I also showed how to run some inline Assembler on the Raspberry Pi Pico 2!

Arduino C++ vs MicroPython Smackdown
youtu.be/MS_BM...

youtube.com
Arduino C++ vs MicroPython Smackdown

07

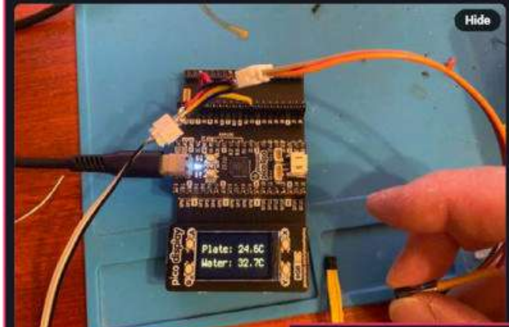
mirobo Technology
@mirobotech

I'm working on a modular beginner Pico board that is able to be assembled and used in multiple configurations, including as a robot!

08

Dr Footleg (he/him)
@drfootleg@fosstodon.org

@themagpi I built a dual temperature sensor Pico (RP2040) based project with circuit python. Ready now for heat pumping experiments with a water cooling rig and Peltier plates. #MagPiMonday

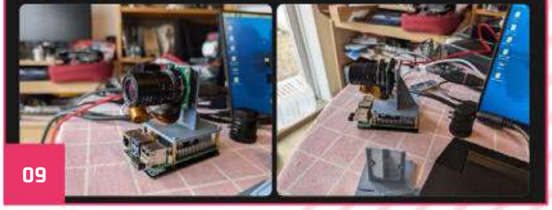


09

briancorteil 37m

Design and 3D printed a mount for the HQ or global shutter Raspberry cameras. You can see my failed first design in the foreground of the second photo.

MagPiMonday



10

Stewart Watkiss
@stewartwatkiss

I've been trying out some GUI programming for the Raspberry Pi.

Part of the challenge was in choosing the GUI framework and I settled on PySide6 based around Qt.

Did I make the right choice?

penguintutor.com/programming/gu...

#MagPiMonday

Tooth timer

Taking a more direct approach to switching lights on and off with Raspberry Pi Pico

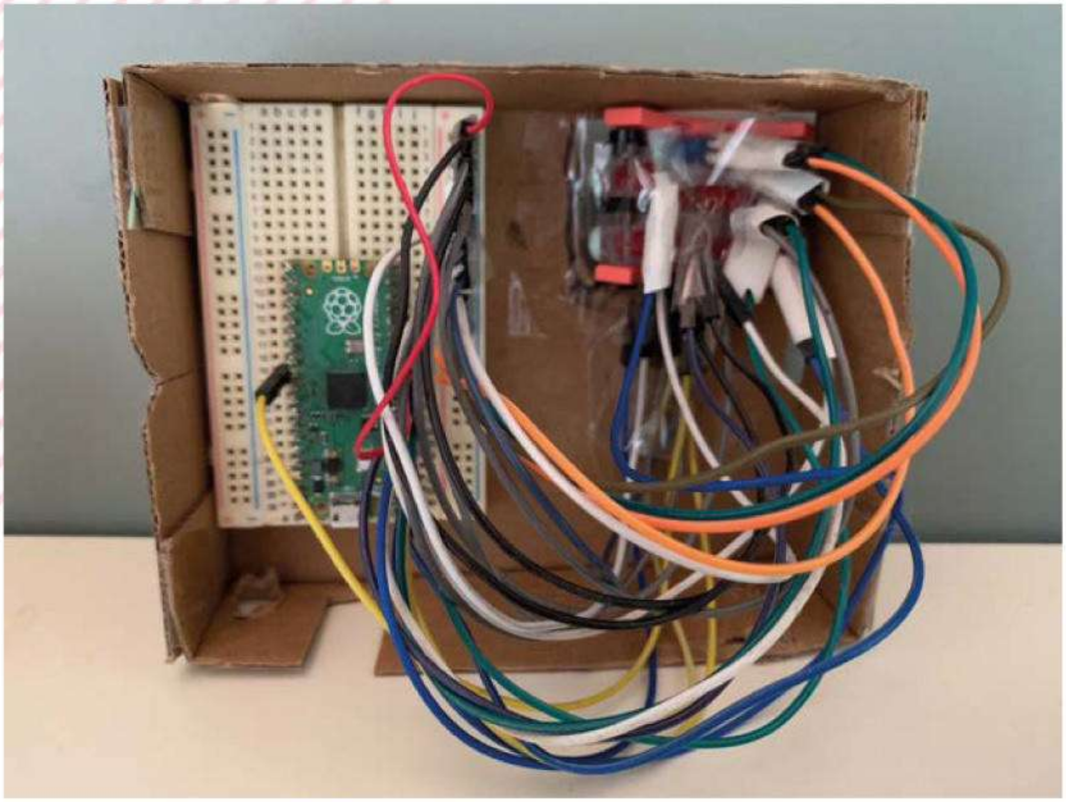
We recently got an email from 11-year-old Siddhan, who is the brother of Nirvaan from last issue, about his LEGO and Raspberry Pi 'tooth timer'.

"It lights up areas of your mouth with some custom LEGO dentures I made to fit LEDs," Siddhan tells us. "The model shows which teeth to brush for how long, and can also tell you if you need to brush the inside or edges of your teeth based on the blue and green lights! When I was young my mum used to tell me where to brush (and it was very helpful) but now she has other things to do so I wanted to recreate this! Before this invention was here, I would get distracted while brushing and would just brush a few teeth, so I then decided to make my model."

It's written in Bipes, which is a block-based programming language, and you can find the code here: **magpi.cc/toothtimer**

When looking for a case, sometimes the nearest cardboard box does the job

Lights show you where to brush, and stay on for the length of the brushing



◀ Tooth Timer uses a breadboard, a Pico, and some block code

Crowdfund this

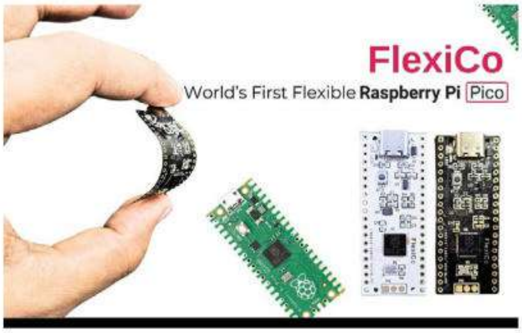
Raspberry Pi projects you can support this month

DigiPort

This Compute Module 4 powered device is billed as your new pocket PC, being able to turn any HDMI screen into a PC by just plugging it in – and adding some USB-C power to it too. With the wireless capabilities of Compute Module 4, you'll also be able to easily add wireless mice and keyboard to the system.

▶ **kck.st/4gLjZOx**

FlexiCo



A very neat concept of a board like a Raspberry Pi Pico that can bend and flex. It has an RP2040 onboard, and uses a USB-C connector as well – otherwise the I/O are just about the same as a regular Pico. It's launching soon on Kickstarter, so look out for it

▶ **magpi.cc/flexico**

Your Letters

Identified flying object

I was very interested in David Miles' joystick article in Issue 146 (magpi.cc/146).

I flew the Panavia Tornado F3 as a Royal Air Force navigator in the 1990s, and also spent time on the aircraft in the service software maintenance team. So, I'm sure that the two joysticks are working replicas of the pilot's control column and the navigator's radar hand controller (used to steer the radar scanner and perform other radar-related functions).

The equipment is clearly not from an actual aircraft, and neither myself nor former colleagues that I've asked are sure where it was used. My best guess is either the crew training simulator or a ground test rig.

Will you please congratulate David on his ingenuity, share this information with him, and tell him that he's welcome to contact me, if he wishes?

Paul via email

There was a huge amount of interest about this project online – not only from long-time flight simulator heads but also from several people who saw David pick them up at EMF to begin with. It's great to get a proper identification of the joysticks now the article is out too!

We got a further follow-up email from Paul that read: "One small addition/alteration, after some input from a former colleague this afternoon: 'Our best guess is that they come from the Microcomputer Air Intercept Trainer (MAIT), which allowed pilots and navigators to train in positioning the aircraft, and the radar, to intercept targets; with everything simulated and displayed using a standard personal computer'."

Anyway, we now expect the second-hand market for old simulator joysticks to skyrocket.

▼ Tornado: a classic jet first introduced in the 1980s

Monthly vs annual

I would prefer to make one yearly contribution rather than monthly contributions and receive free early access to *The MagPi* as a PDF. I don't really want to receive the paper magazine.

How can I do this? The contribution form seems to only work for monthly contributions.

Benn via email

Unfortunately we do not currently offer annual contributions to the magazine, although we're always open to feedback on how folks would like to pay for the early-access digital version.

Back Issues

Issue 145
September 2024
We've got huge news: Raspberry Pi Pico 2 has been revealed to the world! This month, we also welcome HackSpace magazine and its readers to The MagPi magazine, which is now bigger than ever at 132 pages. This expanded issue allows us to include HackSpace's amazing projects and tutorials. Together, The MagPi & HackSpace magazine form the greatest maker magazine printed.
More details
Sold out | Get PDF

Issue 144
August 2024
This month, we inspire you to make that next build even better. Our Essential Electronics guide bridges the gap between code and components. Armed with this month's mag you'll be building amazing electronic gadgets in no time.
More details
Sold out | Get PDF

Issue 143
July 2024
Let's get serious about robotics! In the latest edition of The MagPi magazine, you'll discover sensor-packed rovers, smart walkers, and four-legged robotic dogs that can roll, run and trot around alien planets (or your backyard). So get ready for that job at NASA by building your very own robot explorer.
More details
Sold out | Get PDF

Issue 142
June 2024
Raspberry Pi 5 introduced a PCI-express compatible connection enabling you to connect an M.2 SSD drive for huge file storage and super-speedy data transfer. Alongside a much faster processor, this makes Raspberry Pi 5 the ultimate choice for building a do-it-all media player. In this month's edition of The MagPi, we do exactly that!
More details
Buy now | Get PDF

◄ All our magazines are available as free PDFs three weeks after they're released

Scratch the itch

Code Club children love my Raspberry Pi 3 set-up, mainly purchased at your Leeds pop-up shop. I read the article by Eben in Issue 146 about the smaller memory Raspberry Pi 5 and found it very interesting – but the children always say, 'But will it run Scratch 3?'

And I have to confess my Raspberry Pi 3 won't. I have to resort to a Windows PC. Could someone explain, in an article, what the minimum requirement for working in web-based Scratch would be – so that next time I can say 'Yes!'.

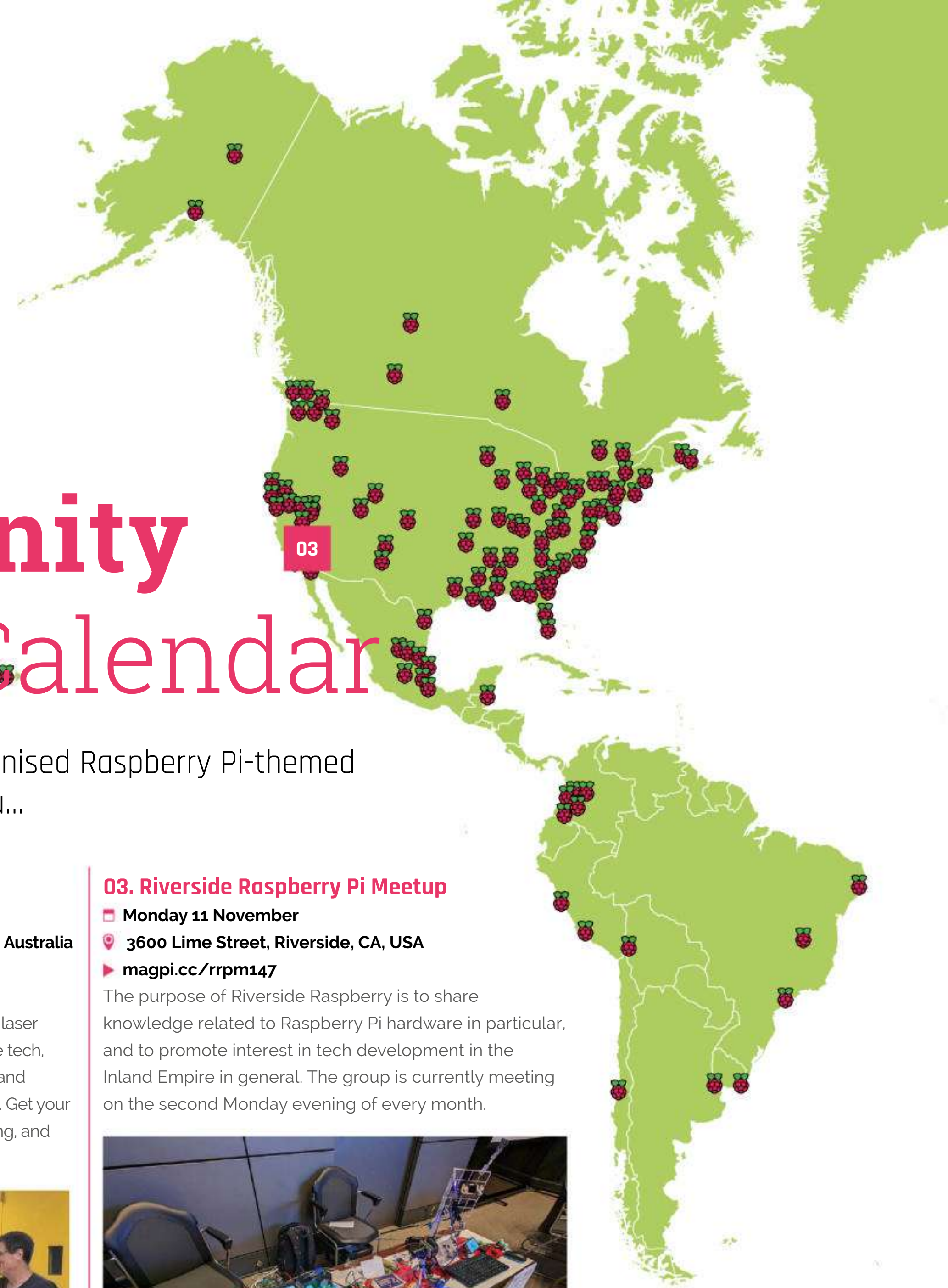
Annes via email

Scratch 3 requires a little more CPU power and memory than a Raspberry Pi 3 can provide – it's actually been available (offline!) on any Raspberry Pi 4 with 2GB or more RAM for a few years now. Memory is the biggest bottleneck, but every Raspberry Pi 5 has a minimum of 2GB of RAM, which means it should run just fine on them.

▲ A nice little setup from Anne using Raspberry Pi products – even the box as a stand

Contact us!

› Mastodon	magpi.cc/mastodon
› Threads	@themagpimag
› Facebook	magpi.cc/facebook
› Email	magpi@raspberrypi.com
› Online	forums.raspberrypi.com



Community Events Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Melbourne Raspberry Pi Meetup

Sunday 3 November

Docklands Makerspace and Library, Melbourne, Australia

magpi.cc/mrpm147

This meetup is open to everyone with an interest in electronics, robotics, home automation, 3D printing, laser cutting, amateur radio, high-altitude balloons, space tech, etc. Makers are invited to bring along their projects and project ideas, and come connect with other makers. Get your questions answered, show off the work you are doing, and get support to resolve nagging issues.

02. PLUG/Perth Open Source Hacking afternoon

Sunday 10 November

Perth Artifactory, Perth, Australia

magpi.cc/plug147

The monthly PLUG + POSH Hack Afternoon is a language/platform-agnostic hack session. Bring your favourite or an interesting project and a love of code, or any other tech project that you're interested in. We'll be hosting it in person at the Artifactory, as well as online.

03. Riverside Raspberry Pi Meetup

Monday 11 November

3600 Lime Street, Riverside, CA, USA

magpi.cc/rrpm147

The purpose of Riverside Raspberry is to share knowledge related to Raspberry Pi hardware in particular, and to promote interest in tech development in the Inland Empire in general. The group is currently meeting on the second Monday evening of every month.

04. Japanese Raspberry Pi Users Group at Ogaki Mini Maker Faire 2024

Saturday 23 November to Sunday 24 November

Softpia Japan Center Building, Ogaki City, Japan

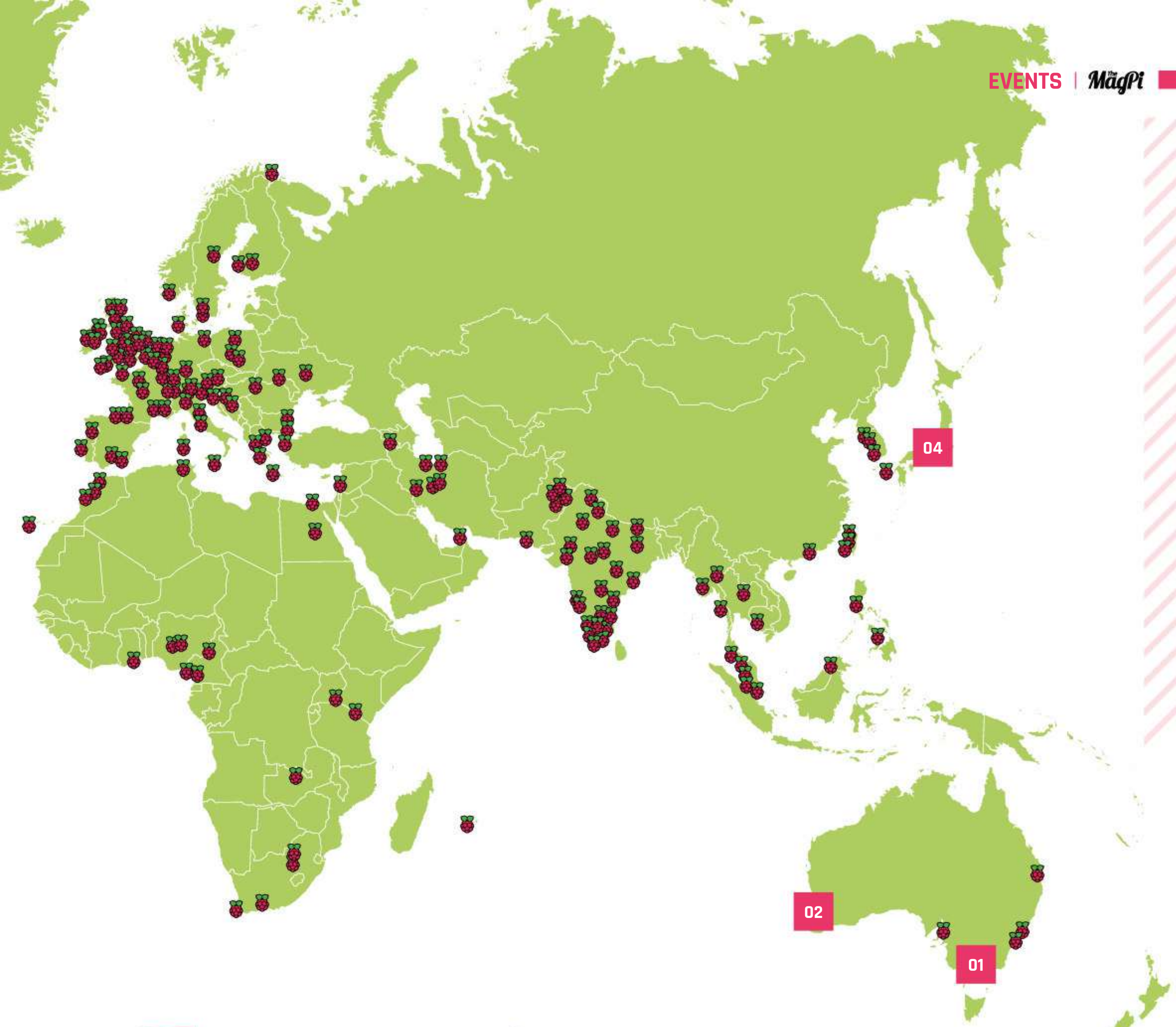
magpi.cc/ogaki24

The Japanese Raspberry Pi Users Group is coming back to the Ogaki MiniMaker Faire in Ogaki, Gifu, which is near to Sony Inazawa, Nagoya. They will exhibit many use cases with Raspberry Pi 5, Pico 2, and more! They'll give away some stickers and more too.

FULL CALENDAR

Get a full list of upcoming community events here:

magpi.cc/events



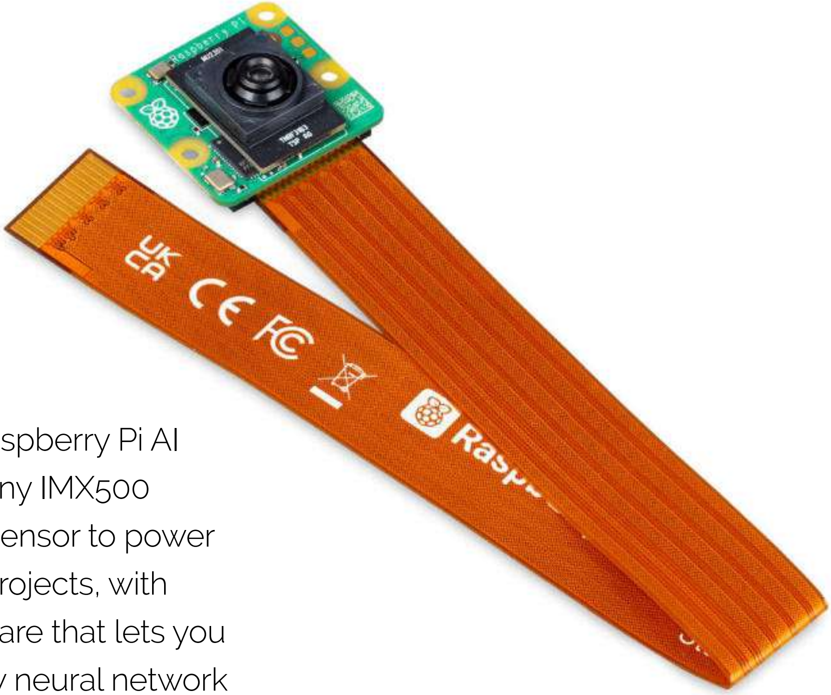
Maker Faire Shenzhen

- Where Vanke Design Commune, Shenzhen, China
- When Saturday 17 November to Sunday 18 November

We're proud to partner with Raspberry Pi Approved Reseller Seeed Studio to be at Maker Faire Shenzhen again in 2024. Come and meet members of the Raspberry Pi team, learn about our latest products (including the Raspberry Pi AI Kit), and share with us the things you've made with Raspberry Pi technology.

magpi.cc/mfs24

WIN ONE OF FIVE RASPBERRY PI AI CAMERAS



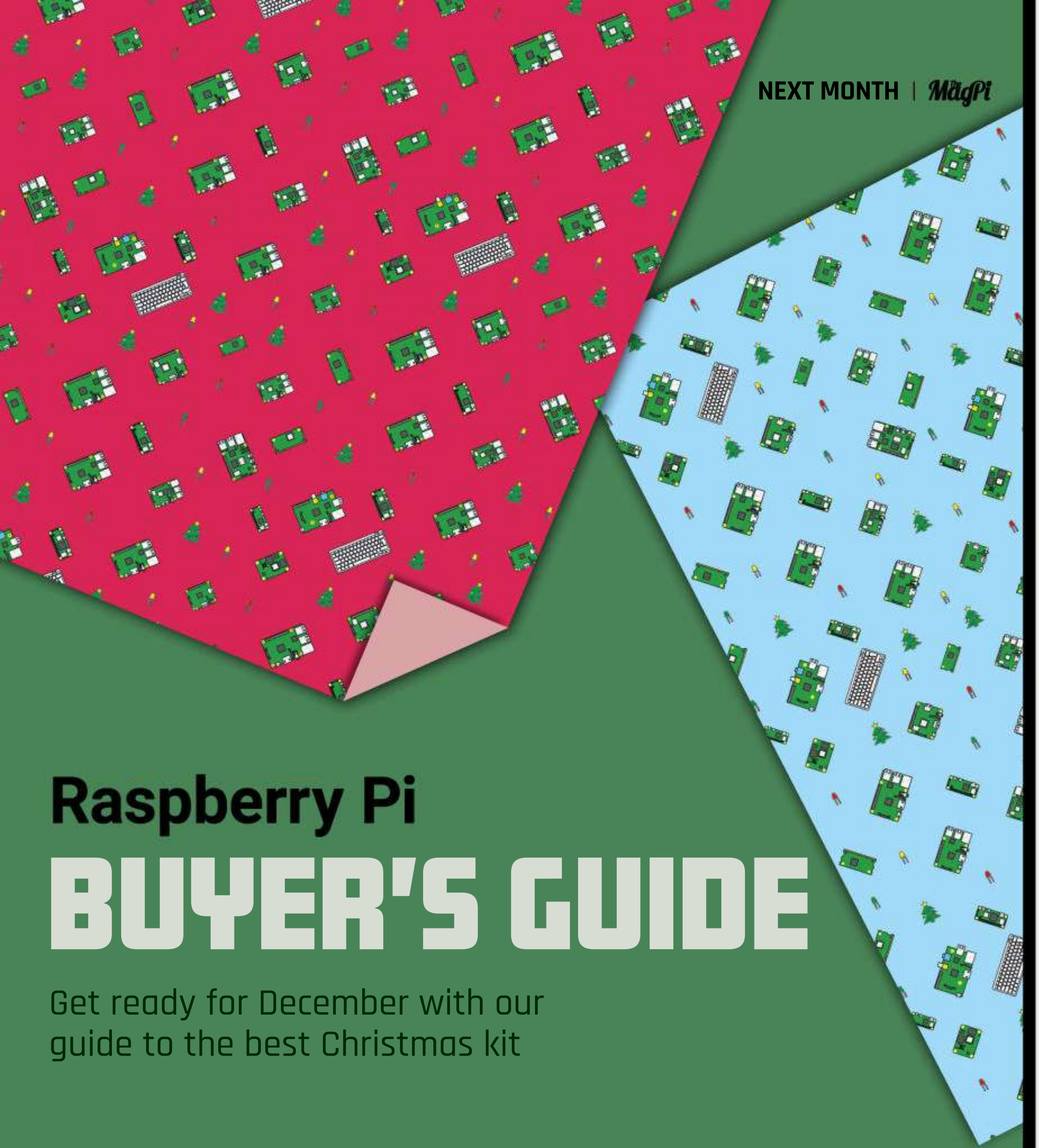
The brand new Raspberry Pi AI Camera uses a Sony IMX500 Intelligent Vision Sensor to power up your AI vision projects, with easy-to-use software that lets you quickly set up new neural network models. We have five to give away.

Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/aicamera

Terms & Conditions

Competition opens on **23 October 2024** and closes on **21 November 2024**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from *The MagPi* magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter (X) or any other companies used to promote the service.

2nd Edition
Updated for Raspberry Pi Pico W
Get started with
MicroPython on Raspberry Pi Pico
The Official Raspberry Pi Pico Guide
PLAYER 1 WINS!
Raspberry Pi PRESS
Gareth Halfacree, Ben Everard



Raspberry Pi
BUYER'S GUIDE

Get ready for December with our guide to the best Christmas kit

The Magpi **#148**
On sale **21 November**

Plus!

Build festive projects

Discover a filament wrapping machine

Tame your own 'catbot'

DON'T MISS OUT! **magpi.cc/subscribe**

MASTODON	magpi.cc/mastodon
THREADS	@themagpimag
FACEBOOK	magpi.cc/facebook
EMAIL	magpi@raspberrypi.com
ONLINE	forums.raspberrypi.com

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Ian Evenden

ADVERTISING
Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

Head of Design
Jack Willis

Designers
Sara Parodi, Natalie Turner

Illustrator
Sam Alder

Photographer
Brian O Halloran

CONTRIBUTORS
Jonathan Bell, Nate Contino, David Crookes, PJ Evans, Jo Hinchcliffe, Rosemary Hattersley, Nicola King, Phil King, Naush Patuck, Andrew Lewis, Rob Miles, Louis Wood

PUBLISHING

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications
Helen Lynn

CEO
Eben Upton

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine.

ISSN: 2051-9982.

Travelling

You can take Raspberry Pi wherever you go – but is there a benefit, **Rob Zwetsloot** ponders?

Next year, I have the feeling I'm going to be doing a lot more long-distance travelling, and that always gets me wondering about how I am going to entertain myself during these journeys. I actually have a pretty good system down – a tablet with books loaded on, a console with a game backlog (sometimes this does mean a 3DS), and hoping there are some cheesy movies I've not seen if I'm flying. I've got a new tablet recently which actually has a decent amount of storage too, so I can finally download shows from streaming services. A revolution.

This recent change got me wondering how else I could mix up the travel tech and, of course Raspberry Pi came to mind. In fact, Raspberry Pi could have solved my issue with not be able to watch my own TV and movies on flights long ago, but thankfully I don't regret not making that connection enough to bother my therapist about it.

Obvious/obscure

Some of my videogame backlog includes a lot of retro games, so the first thought I had was to finally make one of my portable Raspberry Pi game systems a bit more permanent and finally play Japan-only *Treasures of the Rudras* for the SNES (I heard some music for it on a Relaxing SNES Music playlist, sounds fun) among others.

The journey itself isn't the only thing I need to think about though. I could always take a little Raspberry Pi Zero 2 W box that plugs into a hotel TV and VPNs into my home network, so can either stream games or home media wherever I am during any downtime.

> "The first thought I had was to finally make one of my portable Raspberry Pi game systems a bit more permanent"

Maybe I could also have a GPS tracker to truly track my trip from beginning to end! Why? Data is cool, I guess.

Too far

Therein lies the problem – I think putting together a project *just because* will end up being a little too much work before a holiday. And listen, I'm a cosplayer, I know all about undue stress before a weekend off. I've done my time crunching before a trip.

The worst case is that it gets you in trouble too. While writing this, I was reminded of a trip I took to Walt Disney World in 2017. It was a very big deal to me – I won't go into detail here – but I had managed to procure a special board that could interact with signals sent out at the theme parks to co-ordinate LEDs with store window displays or fireworks shows. I had decided to fit mine into a little lightsaber toy, which I had disassembled into its plastic blade and plastic hilt… which was full of wires and batteries. So no wonder I was pulled off to one side once I got off the plane, and had to wait a while for my baggage – the X-ray machine must have seen what looked like a pipe bomb in my bag.

It took me a few days to realise why they had pulled me aside like that, and also that the plastic was too thick for the signals to reach the board anyway. The lesson is: don't do anything too DIY with a Raspberry Pi while you travel. A retro handheld will do just fine.

AUTHOR

Rob Zwetsloot

Rob is Features Editor of *The MagPi* and is unfortunately afflicted with Disney Adult Syndrome.

magpi.cc

PiKVM

Remote control redefined

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

- Power consumption in idle mode: just 2.67 Watts!
- Transfer your mouse and keyboard actions
- Access to all configuration settings like UEFI/BIOS
- Capture video signal up to 1920x1200@60 Hz
- Take full control of a remote PC's power

PiKVM V4 Plus

The most feature-rich edition

- More connectivity
- Extra storage via internal USB3.0
- Upgraded powering options
- More physical security features
- Extra HDMI output
- Advanced cooling solution

A cost-effective solution for data-centers, IT departments or remote machines!

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBII · Cytron Marketplace

ROBU.IN Your ideas, Our Parts · NETTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

No reseller in your country?
Check shop.hipi.io (import fees might apply).

List of official resellers by country: